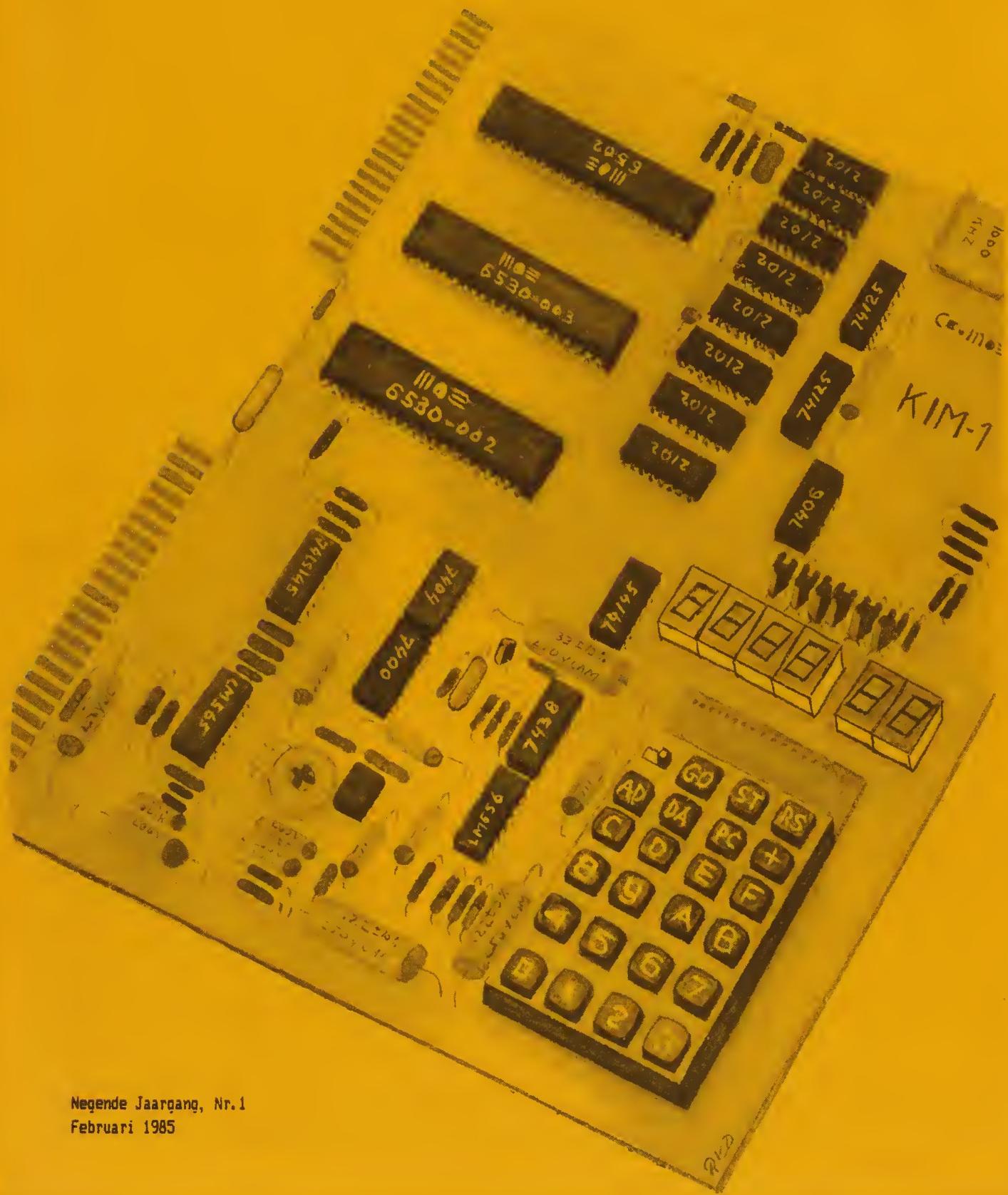


DE 6502 KENNER

36



Negende Jaargang, Nr. 1
Februari 1985

DE 6502 KENNER

** DE 6502 KENNERS ** -- EEN CLUB VOOR 65xx GEBRUIKERS

De vereniging heeft leden in Nederland, Belgie, Duitsland, Frankrijk, Spanje, Portugal, Amerika, India. Het doel van de vereniging is: het bevorderen van de kennisuitwisseling tussen de gebruikers van 65xx-computers, zoals KIM, JUNIOR COMMODORE-64, APPLE, C64, PEARCOM, AIM-65, SYM, PET, BBC ATARI, VIC-20, BASIS 108, PROTON COMPUTERS, ITT 2020, OSI, ACC 8000, ACORN ELECTRON, SYSTEM 65, PC-100, PALLAS, MINTA FORMOSA, ORIC-1, STARLIGHT, CV-777, ESTATE III, SBC 65/68, NCS 6502, KEMPAC System 4, Elektuur SAMSON-65 DOS computer. De kennis uitwisseling wordt o.a. gerealiseerd door 5 maal per jaar DE 6502 KENNER te publiceren (1984 en 1985 6 maal zonder contributieverhoging), door het houden van clubbijeenkomsten, door een cassette-bibliotheek in stand te houden en door het verlenen van daerware-service. De leden zelf organiseren regio-bijeenkomsten.

Verschijningsdata

DE 6502 KENNER

derde zaterdag
van de maanden:
februari, april, juni,
augustus, oktober, december.

Inlichtingen over de regio- bijeenkomsten:

Gerard van Roekel,
Van der Palmstraat 11 - C
3135 LK Vlaardingen.
Tel.: 010 - 351101

De vereniging is volledig onafhankelijk, is statutair opgericht en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305.

Voorzitter:
Rinus Vleesch-Dubois
Fl. Nightingalestraat 212
2037 NG Haarlem
Tel.: 023 - 330993

Penningmeester:
John F. van Soriano
Tulip 71
2925 EW Krimozen/IJssel.
Tel.: 01807 - 20589

Leden:
Adri Hankel (05490 - 51151) Hardware/software
Jan Tomoot (070 - 931417) Promotie
Erwin Visschedijk (05490 - 71416) Hardware/software
Nico de Vries (010 - 502239)
Erevoorzitter: Sieg de Vries
Ereleden : Mw. H. de Vries - Van der Winden
Anton Mueller
Lidmaatschap : Fl. 45,- per kalenderjaar, oostrekening
3757649 t.n.v. KIM Gebruikers Club Ned.
te Krimozen a.d. IJssel.
Advertenties : Tarieven op aanvraag bij de redactie.

Bijeenkomsten van de club

derde zaterdag
van de maanden:
januari, maart, mei,
september, november.

Redactie-adres en informatie over daerware etc.:

Willem L. van Pelt
Jacob Jordaanstraat 15
2923 CK Krimozen/IJssel.
Tel.: 01807 - 19881

Sekretaris:
Gert Klein
Diedenweg 119
6706 CM Wageningen
Tel.: 08370 - 23646

Redactie DE 6502 KENNER:
Willem L. van Pelt
Jacob Jordaanstraat 15
2923 CK Krimozen/IJssel.
Tel.: 01807 - 19881

** DE 6502 KENNER ** -- EEN BLAD VOOR 65xx GEBRUIKERS

DE 6502 KENNER is een uitgave van de KIM Gebruikers Club Nederland. Het blad wordt verstrekt aan leden van de club. DE 6502 KENNER wordt van copy voorzien door leden van de club, bij de omaak van een publikatie bijgestaan door de redactie. De inzendingen van programma's dienen voorzien te zijn van commentaar in de listinos. Dit, in verband met het educatieve aspect. De inzendingen kunnen geschieden in machinetaal, de assembler/disassembler/editor Micro-ADE, de Format Lister/Assembler/Tape-utilities/Editor FATE, de resident assembler/editor van Carl Moser, in de hogere programmeertalen zoals bijvoorbeeld Basic, Basicode, Fort, Focal, Comal, Pascal, etc. etc. De leden schrijven ook artikelen over de door hen ontwikkelde hardware en/of aanpassingen daarop. Zij schrijven tevens artikelen van algemene aard of reageren op gepubliceerde artikelen.

DE 6502 KENNER is een blad van de leden en door de leden.

De Micro-ADE is een product van Micro-Ware Ltd., oescnreven door Peter Jennings en bestemd voor alle 6502-computers. De Kim Gebruikers Club Nederland heeft de copyrights verworven nadat Sebo Woldringh de oorspronkelijke 4K KIM-1 versie met 4K uitbreidde tot 8K KIM-1 versie, welke later door Adri Hankel werd aangepast voor de JUNIOR. Uit de berg aanvullingen en wijzigingen stelde Willem van Pelt een nieuwe 8K source-listino voor de JUNIOR samen. De implementatie op andere systemen dan de KIM-1 en de JUNIOR kan eenvoudig gebeuren door het aanpassen van de I/O-adressen, welke in de source-listino duidelijk terp te vinden zijn. Vraag de orijslijst bij de redactie door het zenden van een gefrankeerde (fl.1.40) retourenveloppe.

De Format Lister/Assembler/Tape-utilities/Editor FATE staat dankzij de medewerking van Proton Electronics te Naarden nu voor de bezitters van een JUNIOR-computer ter beschikking. Het 12K programma vindt haar oorsprong in het universele disk operatino systeem van Proton Electronics. Rob Banen publiceerde daarover enige wijzigingen en schreef daarop FATE. Eenmaal met FATE bekend, dan is de stap naar het door ons gewaardeerde disk operatino systeem nog slechts een kwestie van geld. FATE wordt beschikbaar gesteld met toestemming van Proton Electronics te Naarden.

In de edities van DE 6502 KENNER worden regelmatig aankondigingen gedaan van de door de club georganiseerde landelijke en regionale bijeenkomsten. Ook worden bestuurlijke mededelingen gedaan, naast informatie over hetoeen op de markt te kooi is. Leden kunnen het gevraagd of te kooi aangeboden artikel bekend maken of brieven aan de redactie laten publiceren.

Elke editie van DE 6502 KENNER omvat minimaal 48 pagina's.

De redactie verwacht inzendingen van gebruikers van alle 6502 systemen.

Het publiceren van artikelen wordt op de voet gevolgd door een onafhankelijke jury van 6502 kenners bij uitstek. Deze jury kent jaarlijks een aantal aanmoedigingspremies toe voor de beste publikatie in DE 6502 KENNER.

DE 6502 KENNER is een uitgave van de KIM Gebruikers Club Nederland.

Adres voor het inzenden van en reakties op artikelen voor DE 6502 KENNER:
Willem L. van Pelt
Jacob Jordensstraat 15
2923 CK Krimpen/IJssel
Tel.: 01807 - 19881

Vaste medewerkers:
Willem van Pelt
Gerard van Roekel
Frans Smeehuizen
Jaap van Toledo

Freelance medewerkers:
Frans Bakx
Rob Banen
Fridus Jonkman
Gert Klein
Roder Langeveld
Anton Mueller
Gert van Dobroek
Ruud Uphoff

Illustraties:
Jack Uchtmann

Gehalte of gedeeltelijke overname van de inhoud van DE 6502 KENNER zonder toestemming van het bestuur is verboden. Toedassing van gedubbelde programma's, hardware etc. is alleen toegestaan voor persoonlijk gebruik.

DE 6502 KENNER verschijnt 6 x per jaar en heeft een oplage van 500 exemplaren.

Copyright (C) 1985 KIM Gebruikers Club Nederland.

De voorpagina is een aquarel van een KIM, geschilderd door:
Rinus Vleesch Dubois.

In verband met auteurswetgeving en andere maatregelen op het gebied van bescherming van software kan de redactie geen aansprakelijkheid aanvaarden voor inzendingen.

Inzendingen dienen afkomstig te zijn van de inzender, tenzij anders aangegeven.

INHOUDSOPGAVE DE 6502 KENNER NR. 36 FEBRUARI 1985

1. Van de redactie	2.
2. CAT, Club Assembler Texteditor Conditionele Assembler voor alle 6502 computers ... Ruud Uphoff	3.
3. UITNODIGING Bijeenkomst Geldrop 16 maart 1985	4.
4. JUNIOR met VDU-kaart Het Grafisch Display ... J.J.A. en J.A.J. Janssen GREEDY. Deel 2. Soeloprogramma in 3 delen. ... Phons Bloemen	5.
5. JUNIOR Coordinateen bij USURPATOR II schaakoprogramma ... Gamiel de Vy, België Large Characters with the Microline 80 ... Frans Bakx	23.
6. 65C816 De 65SC816. een 16-bits ovolger voor de 6502 ... Gert Klein	41.
7. BASIC Tokenized Microsoft Basic Keywords and Addresses C-16 ... Nico de Vries Programma Uitnodiging tot ... Alfons van de Meutter. België Information about OSI nine digit Basic ... Gert Klein	46.
8. COMMODORE Monitor Programma's voor Commodore Computers ... Nico de Vries	13.
9. FORTH How to change the Memory Map of your FORTH system on JUNIOR ... Gert van Dobroek FORTH op JUNIOR computers. Deel 1 ... Gert van Dobroek	37.
10. HARDWARE Het Grafisch Display (voor JUNIOR met VDU-kaart) ... J.J.A. en J.A.J. Janssen	21.
11. Micro-ADB Inlezen source	5.
12. OHIO-DOS V3.3 Aanpassen I/O-routines	48.
13. BBC BBC microcomputer LEDS aan Userpoort ... Jean van Huffel. België	24.
Acorn Electron ... Simon Voortman	40.
14. MARKTINFO Flitsend FORTH. Boekbespreking ... Peter van Harten AGENDA Radio Den Haag/Rotterdam/Leiden e.o. Vraag en Aanbod Verkrijgbaarheid verschenen edities	45.
	24. 31. 32. 44. 45.
	20.

Een nieuw computeriaar is ingedaan. Een jaar waarin nieuwe computers het daallicht zullen aanschouwen. Een jaar waarin we ons nieuw verheerlijkt het beeld voor ons zien van die zo eindeloos bereikbare. o zo begeerde computer met een maximum aan RAM-geheugen, een minimum aan storinoen, een computer waar zelfs je vrienden de lippen bij af zullen likken, een computer die werkelijk alles kan met uitzondering van die kleinigheid waar je vrouw zo naar uitkijkt, een gewilde computer omdat 'ie je alle werk uit handen neemt, omdat 'ie niet alleen Basic kent, maar ook met assembler weet om te gaan, en met FORTH natuurlijk, en met LOGO, en met PASCAL, en met alles wat er nog uitgevonden moet worden, een computer die zo gedroptrogrammeert is dat hij vreselijk interactief met je staat te communiceren, kortom, een computer die je alleen in je fantasie kunt ooslaan omdat hij in werkelijkheid nooit tot je eigendom kan worden vanwege de extreem hoge kosten.

Zijn we met beide benen op aarde teruggekeerd, dan kunnen we nuchter vaststellen dat we het over het algemeen doen met primitieve computers vergelijken met onze fantasiewerkerkheid, maar dan op zo coed modelijke manier. Dat is wat onze club bedoelt. Een club met mensen die de beoefening van het hobbyoeuvre kennen, maar het desalniettemin volledig benutten.

Onze club verheugt zich erop een reeks nieuwe leden te mogen begroeten. We hopen dat zij zich binnen onze gelederen thuis zullen voelen. Enkele omerkenden zijn te plaatsen. In verband met een wijziging in de begeleiding van nieuwe leden, worden deze in de eerste plaats geregistreerd bij de redactie, daarna bij de penningmeester (voor de te ontvangen lidmaatschaarden), en dan bij de sekretaris (die voor de toezending van DE 6502 KENNER zorgdraagt). Al bij registratie op de redactie kunnen nieuwe leden nog voor de verschijning van een editie op de hoogte worden gebracht van bijeenkomsten, welke zij anders zouden hebben moeten missen. De nieuwe leden kunnen meteen worden geïnformeerd over de relatie welke er bestaat tussen het lidmaatschap en het verschijnen van DE 6502 KENNER. We kunnen directer benadrukken dat DE 6502 KENNER een blad is dat niet wordt gevuld door redactie- of bestuursleden, maar dat het door de leden zelf dient te worden volgeschreven. Waarbij boven dien aan de orde kan komen dat we net zo blij zijn met programma's in een of andere hogere programmatuur als in assembler, ook wel machinetaal. We zullen niet nalaten gebruikers van alle soorten computers in de 65XX-familie op te wekken voor al hun programma's, artikelen en andere vormen van kennis toe te voeden aan de coov-buffer. Dit alles onder het motto dat het publiceren van eigen materiaal anderen opeukt ook tot een publicatie over te gaan.

De redactie, die naast activiteiten als cassette-bibliotheek en paperware-service ook nieuwe leden werft en aanspoedt deze te werven, geeft ondersteuning aan landelijke en regionale bijeenkomsten, informeert de leden over hetgeen er verkrijgbaar is en op welke wijze kan worden gezocht naar het oplossen van problemen, waarbij dikwils contact gezocht wordt met andere leden. Hoewel de edities zich bij uitstek lenen voor het doorgeven van informatie over clubfaciliteiten, er wordt slechts bescheiden gebruik van gemaakt, om zoveel mogelijk ruimte te geven aan de inzendingen van de leden.

Op deze plaats en op dit tijdstip lijkt het goed een reeks zaken nog eens de revue te laten passeren.

In principe verstrekt de club voor fl. 45,- lidmaatschap vijf edities van DE 6502 KENNER. Enige tijd geleden was de omvang daarvan 40 pagina's, terwijl we nu minimaal 48 pagina's brengen, voor diezelfde fl. 45,-. En dat is niet alles, want de informatiedichtheid is zo ongeveer verdubbeld. Er staat dus veel meer in. Dat is echter niet alles. Uit de verkopen van cassettes en paperware houdt de club na aftrek van alle kosten, een kleinheid over. De service bleek echter dermate oorzaak, dat het moeilijk het bestuur in 1984 en in 1985 de uitgave van een gratis zesde editie te realiseren. Met het daarenboven ook nog omgaan brengen van de entree van landelijke bijeenkomsten van fl. 25,- naar fl. 10,- denk ik dat we de oren van het modelijk bereikt hebben.

In 1984 hebben we getracht duidelijk te maken dat alle leden best een steentje bij kunnen dragen aan de ledenvervulling. In een enkel geval bracht een lid zelfs meer leden binnen. Proficiat! Als we alle activiteiten meten, dan kan ik uitspreken dat het niet onderdeel voor andere wervingen. Om een perspectief te realiseren waarin een blijvende garantie voor een zesde editie staat, zullen we met z'n allen nog meer de schouders eronder moeten zetten.

Het uitbrengen van de in Nederland en Belgie uniek genoemde edities van DE 6502 KENNER zou niet goed modelijk zijn als de leden niet zouden beseffen dat hun lidmaatschap met zich brengt dat juist zelf het blad moeten vullen. Dat dat in de praktijk betekent dat de leden hun inzendingen toevoegen aan de coov-buffer en dat de redactieveradering daaruit zo optimaal modelijke samenstellingen vaststelt, staat het principe niet in de weg. De hoeveelheid inzendingen is nog niet gevarieerd genoeg. Hoewel we een aantal schitterende hardware-publicaties ontvingen, het zijn er nog te weinig. De Commodore 64 en de Commodore 16, laatstgenoemde met zijn veel omvanger Microsoft Basic, is te veel nog afwezig. Reden dus om de bezitters op te wekken hierin verandering te brengen. Datzelfde geldt eveneens de BBC, de Amstrad en andere. Toonaangevend is wel de Elektuur zelfbouwcomputer JUNIOR. Die schare van inzenders hebben niet nadelen steeds aan DE 6502 KENNER te denken. De lof welke op de ledenveradering aan de redactie werd toedezwaaid komt de inzenders geneel toe.

Vroeger ontving de redactie hexdummys en niet-bemerkarieerde listings. Het is alweer enige tijd geleden dat we oesloten de inzenders te vragen vollediger te zijn. Inzendingen moeten voorzien zijn van auteursnaam, korte inleiding, en van commentaar.

Met nadruk vraag ik na te gaan of een programma ook een eenvoudige wijze valt aan te passen aan andere machines. In de voorstaande aardlagen hebben enkele auteurs niet onbelangrijke aanwijzingen gedaan in oopubliceerde programma's.

Het valt mij op dat sommigen de waarden van een eenvoudige aanpassingen of hele kleine programma's onderschatte. Ach, zeet men, daar kan toch iedereen wel ookomen? Of dat is toch niet interessant genoeg? In zulke gevallen heb ik stijf en stram volgehouden dat het onjuist is zo te denken. De redactie heeft immers andere ervaringen! Als U zich eigen maakt alle zelf gemaakte programma's, klein of groot, in te zenden, dan voorspel ik 1985 bij voorbaat tot een voor DE 6502 KENNER geslaagd jaar!

W.L. van Pelt.

*** *** *** *** *** *** *** *** *** ***
*** C A T *** *** *** *** *** *** *** ***
*** *** *** *** *** *** *** *** *** ***

(c) Ruud Uphoff

CAT wil niets anders zeggen dan: "Club Assembler Teksteditor" Het is een 8k "two pass" assembler zoals b.v. ASSM/TED van C.Moser, waarmee ik tot nu toe altijd heb gewerkt. De oer-versie van CAT is dan ook m.b.v. van ASSM/TED geschreven, totdat CAT het, zij het gebrekig, alleen af kon en vanaf dat moment kon worden gebruikt om zichzelf uit te werken tot het product dat ik U nu kan aanbieden. De teksteditor vertoont uiterlijk een grote overeenkomst met ASSM/TED maar heeft hier en daar andere namen voor commando's. Voornamelijk een kwestie van persoonlijke smaak. Commando's zoals MOVE, COPY, DELETE en RENUMBER. Dit laatste commando kan ook een deel van de source hernoemen. Bijzonderheid in het commando CHANGE is dat het speciale faciliteiten heeft voor het begin en einde van de regel en het onderscheid kent tussen instructie en commentaar op een regel. Bovendien kan gebruik gemaakt worden van een "wildcard" karakter dat in de vervangende string kan worden aangehaald ! De assembler is geheel anders van opzet dan ASSM/TED. CAT kent geen macro's, conditional assembly of relocateable object files. Daarvoor in de plaats is CAT voorzien van ingebouwde structuurmacro's die het de programmeur mogelijk maken om, maar dan wel op assembler niveau, te werken met controlstructuren zoals in een hogere programmeertaal. Gezien mijn voorkeur voor COMAL is het dan ook niet verwonderlijk dat we IF-THEN-ELSE, REPEAT-UNTIL, WHILE-DO en CASE-WHEN-OTHEWISE tegen komen. Vanzelfsprekend is CAT geschikt voor de C-MOS processors, inclusief die van ROCKWELL. Een andere bijzonderheid is dat CAT absoluut niet systeem afhankelijk is. De I/O moet namelijk voor elk systeem apart worden geschreven en daartoe is CAT voorzien van een aantal standaard variabelen en een JMP-tabel. Dat wil niet zeggen dat U altijd eerst dagenlang moet zweegen om CAT op uw systeem aan de praat te krijgen want voor de bekendste machines is of wordt, dat al gedaan. Op dit moment is CAT kant en klaar in de cassettebibliotheek voor de COMMODORE-64. Op het moment dat U dit leest zijn waarschijnlijk ook de JUNIOR en APPLE versies gereed. Voor andere machines kunt U contact met mij opnemen. CAT is tegen de gebruikelijke onkostenvergoeding bij de redactie verkrijgbaar en wordt geleverd met een Nederlandstalige handleiding. Tenslotte zult U mij willen verschonen als ik nog even een paar dingen zakelijk aan de orde stel:

- Conform de statuten blijft CAT eigendom van de auteur: Ruud Uphoff.
- CAT wordt met zijn goedvinden, uitsluitend via de redactie van de 6502-KENNER, gratis ter beschikking gesteld, uitsluitend aan leden van de KIM GEBRUIKERSCLUB NEDERLAND en uitsluitend voor persoonlijk gebruik.
- De vereniging is gerechtigd de voor de verstrekking te maken onkosten in rekening te brengen.

VOORKOM a.u.b. DAT ONZE CLUB SOFTWARE IN DE "HANDEL" BELANDT

DE 6502 KENNER

UITNODIGING BIJENKOMST

Datum : zaterdag 16 MAART 1985
 Lokatie : Gemeenschaoshuis "De Zes Gehuchten".
 Padenvoort 10 te Geldrop (bij Eindhoven) Zie voor entreeoriëntatie
 onderstaand programma.

Reisroute :

- per auto - vanaf snelweg E3 Eindhoven - Venlo
 Afslag Geldrop bij Golden Tulip hotel.
 Bij eerste verkeerslicht richting Centrum linksaf slaan (Emopad). Bij Kruijsinga na ca. 800 m. linksaf slaan (Hoofd Geldrop). Deze weg blijven volgen tot gebouw naast kerk aan rechterzijde.
- vanuit rondweg Eindhoven
 Afslag Geldrop bij DAF showroom (Eindhovense weg). Weg volgen tot tweede verkeerslicht voor soorwegviaduct. Rechtsaf slaan en meest rechtse weg inslaan (Padenvoort). Deze weg blijven volgen tot gebouw voor kerk aan linkerzijde.
- vanuit Nuenen - Helmond
 In Geldrop richting Eindhoven. Bij eerste verkeerslicht voorbij soorwegviaduct linksaf (voorsorteren). Verder als bijroute vanuit rondweg Eindhoven.
- per trein - Stoptrein Eindhoven - Weert
 In Geldrop direct na uitgang station links voetgangerstunnel door (Tournooiveld). Weg oversteken en rechtsaf gaan. Eerste laan links (Hertogenlaan). Na bocht eerste straat rechts (Gildestr.). Eerste straat links (Dieenvaart). Gebouw aan einde links (naast kerk van Zesgehuchten).

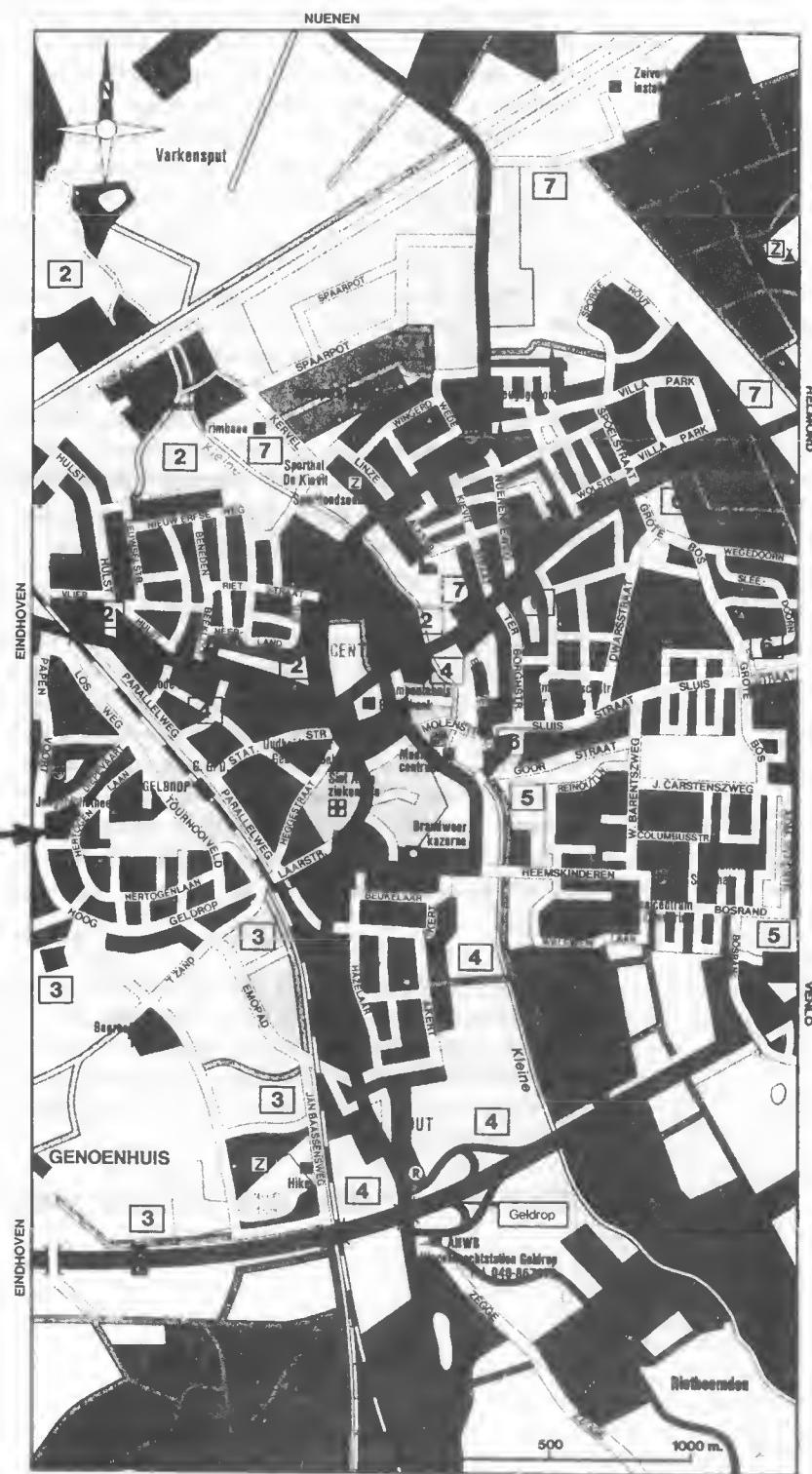
Lunchoakket zelf meenemen.
 Consumpties tegen betaling.

Programma Buitengewone Ledenvergadering: (alleen dit gedeelte gratis toegankelijk)

- 09.30 - Zaal open.
 10.15 - Financieel jaarverslag 1984 en
 Begeleiding 1985.

Programma Bijeenkomst Geldrop (entree: Fl. 10,00 per deelnemer)

- 10.45 - Opening/Lezing Adri Hankel en Erwin Vischedijk over DOS 65 van Ad Brouwer.
 11.45 - Forum.
 12.15 - Lunch.
 13.00 - Markt.
 13.30 - INFORMEEL GEDEELTE.
 Het informeel gedeelte stelt de deelnemers in de gelegenheid onderling met elkaar kennis te maken. BRENG OOK UW SYSTEEM MEE!!!!
 17.00 Sluiting.



HET GRAFISCH DISPLAY

De VDU-kaart van ELEKTUUR heeft als nadeel dat grafisch werken (plotten van grafieken e.d.) niet erg goed gaat. Daarom hebben wij een grafisch display gemaakt, dat in combinatie met de VDU-kaart werkt. Het grafisch display bevat, grofweg gezegd, alleen maar geheugen en stuurologica en is ooggebouwd op een eurokaart m.b.v. het zogenaamde 'road-runner' principe. Deze eurokaart is boven op de VDU-kaart geplaatst waarbij aan de lange zijde van de beide printen de onderlinge verbindingen ontstaan.

FORMAAT

Het beeldformaat van het grafisch display is afh. van de beeldindeling van de VDU-kaart en de hoeveelheid aanwezige RAM-geheugen. Bij aanname van een volledige RAM-bezetting (16Kbyte groot), 2 kleuren en een beeldindeling van 80 karakters op 25 regels geldt voor het formaat:

$$8 * 80 = 640 \text{ puntjes horizontaal}$$
$$8 * 25 = 200 \text{ puntjes verticaal}$$

(128000 van de $16 * 1024 * 8 = 131072$ puntjes worden hier gebruikt.)

Bij een beeldindeling van 64 karakters op 32 regels wordt het formaat:

$$8 * 64 = 512 \text{ puntjes horizontaal}$$
$$8 * 32 = 256 \text{ puntjes verticaal}$$

(Geheugen wordt nu volledig gebruikt)

Zo is elk formaat software-matig in te stellen, maar voor de eenvoud gaan we steeds uit van het formaat 640 * 200 ook de software werkt met dit formaat.

De grootte van de puntjes, bij 2 kleuren, is gelijk aan die van de VDU-kaart ongeacht het formaat en hoeveelheid aanwezige RAM-geheugen. Bij meerdere kleuren (grijstinten) en in de knippermodus neemt de resolutie af, maar daar later meer over.

Het beeld is als volgt ooggebouwd:

Elke geheugendaats (1 byte) levert 8 puntjes horizontaal en 8 opeenvolgende geheugendaats leveren 8 rijen van 8 puntjes.

De 9de geheugendaats is weer het begin van een nieuwe 8*8 matrix naast de eerst etc.

Totdat de 80 kolommen vol zijn waarna de nieuwe matrix onder de 1ste komt etc. Zie figuur 1.

HET RAM-GEHEUGEN

Voor het RAM-geheugen is gebruik gemaakt van de 6116 (150nS) of aanverwanten. De 6116 is 2K*8 bits groot en er kunnen er max. 8 gebruikt worden.

Omdat 16Kbyte in het geheel te veel ruimte in beslag neemt is het geheugen opgedeeld in 2 banken van elk 8Kbyte groot.

Deze banken staan op \$B000 t/m \$CFFF, welke bank geselecteerd wordt bedoelt de PIA (6821 zie schema IC 16) die ook in het grafisch display aanwezig is. Buiten de ruimte besparing levert dit ook interessante mogelijkheden voor latere uitbreidingen.

Interessant is het om te bekijken of de statische RAM's niet vervangen kunnen worden door de veel goedkopere dynamische RAM's.

Echter de 6116 gaf al enige problemen a.g.v. timinafoutjes, daarover meer bij het bespreken van het schema, zodat het gebruik van dynamisch RAM niet simool zal zijn maar theoretisch wel uitvoerbaar.

KLEUREN en KNIPPEREN

Standaard is het mogelijk om met grijstinten te werken, alhoewel de koozelijn met de VDU-kaart zo goedkoop mogelijk is gehouden is het resultaat goed. Indien gebruik wordt gemaakt v/d kleurenmodulator van ELEKTUUR zal het resultaat beter zijn en de grijstinten worden dan kleuren. Omdat dit een dure oplossing is worden er alleen grijstinten gebruikt.

Er zijn 3 kleurmodes:

- A) Geen oriëntinten
 -Max. resolutie
 -Elk bitje stelt een puntje voor. indien bitje is '1' dan een wit puntje anders een zwart puntje.
- B) 3 kleuren: wit, grijs en zwart
 -2 bitjes vormen samen een puntje
 -resolutie 2* zo klein als bij A).
 - b1 b0 kleur
 0 0 puntje niet aanwezig dus zwart
 0 1 idem
 1 0 wit
 1 1 grijs
- C) 8 kleuren: wit, 6 oriëntinten en zwart
 -4 bitjes vormen samen een puntje
 -resolutie 4* zo klein als bij A).
 -Het MSB-bitje, hier dus b3, bepaalt ook weer of er een puntje aanwezig is (indien '1') of niet. Terwijl b0, b1 en b2 de kleuren bepalen (000=wit en 111=zwart).

Zoals te zien is bepaalt het MSB-bit (dus b1 bij 3 kleuren en b3 bij 8 kleuren) of het puntje aanwezig is en de andere bit(s) de kleur. Hierdoor is het mogelijk om eerst overal de kleuren in te vullen en daarna pas het puntje te laten verschijnen d.m.v. het MSB-bitje te setten.

Voor knipperen moet het display in de kleurmode staan en Pa5=1, bij 3 kleuren vervalt de oriëntint en bij 8 kleuren blijven er nog 4 kleuren over.

Het MSB-bitje bepaalt weer of het puntje aanwezig is (indien '1') het LSB-bitje (meest rechtse bitje) bepaalt het volgende:

- Indien '0' -Geen knipperfunktie
 indien '1' -Knipperfunktie, het desbetreffende puntje kan men nu laten knipperen en wel op twee manieren:
 A) Hardware matroso dmv de aanwezige oscillator (Pa6='1')
 B) Software matroso dmv Pa7 (indien Pa6='0')

Pa6 bepaalt dus met welke manier er geknipperd wordt.

Ook hierbij geldt weer dat zolang het MSB-bitje='0' het puntje niet aanwezig is ongeacht de waarde van het LSB-bitje.

DE PIA

De PIA is geselecteerd van \$1980 t/m \$19FF. dit is alleen mogelijk als de VIA (of de interface kaart) op \$1800 t/m \$18FF staat. Hoe dit gedaan moet worden is al meerdere malen aangegeven in dit blad.

Van de soorten worden alleen Pa0 t/m Pa7 en Pb7 gebruikt en wel voor het volgende:

Pa7	Pa6	Pa5	Pa4	Pa3	Pa2	Pa1	Pa0	FUNKTIE
							0 0	eerste 8K RAM
							0 1	tweede 8K RAM
							1 0	voor evt. RAM uitbreidings 1 1 idem
							0 0	2 kleuren
							0 1	3 kleuren
							1 0	8 kleuren
							1 1	display uit
							0	inverse beeld
							1	normaal beeld
							0	normaal
							1	knipper mode
							0	knipperen mbv. Pa7
							1	knipperen mbv. de oscillator
							0	'knipper-puntje' aanwezig
							1	'knipper-puntje' niet aanwezig

Zoals te zien is moeten Pa0 t/m Pa7 allen als uitoang geschakeld worden.

Pb7 wordt als ingang geschakeld en wordt verbonden met de synchronisatie signalen.

Indien de software nu steeds wacht totdat Pb7='0' voordat het geheugen gebruikt wordt, dan zal er geen storing op het beeld komen. Dif gaaf wel ten koste van de snelheid, waar snelheid gewenst is kan het niveau van Pb7 genegeerd worden.

HET SCHEMA

De 4 selectors (IC10 t/m IC13) bepalen of de VDU-controller of de uP bij het geheugen komt.

Om timingfoutjes te ontlopen moeten deze van het type 74S157 of 74ALS157 zijn. Om dezelfde reden is IC32 toegevoegd, deze zorgt ervoor dat het r/w signaal (oen 21 van IC1 t/m IC8) pas dan laag wordt als alle signalen stabiel zijn en weer hoog wordt voordat de VDU-controller het geheugen weer overneemt. De gebruikte waarden van de R's en C's gelden voor de 6116 150nS en moeten bij langzamere typen waarschijnlijk aangepast worden.

Merk op dat van de VDU-kaart een adreslijn (MA9) geïnverteerd moet worden dit kan gedaan worden door N15 op de VDU-kaart, daar deze nu niet gebruikt wordt. De 8 uit 3 decoder (IC9) bepaalt welke RAM geselecteerd wordt.

IC31 buffert de data die uit het RAM-geheugen komt, achteraf gezien is dit niet nodig en kan evt. weggelaten worden, waarbij dan de ingangen met de uitoingen doorverbonden moeten worden.

IC27 en IC28 zorgen ervoor dat afh. van de kleurmode de data v/h schuifregister (IC29 en IC30) op de juiste tijd ingelezen wordt in IC26.

4Q van IC26 bepaalt of er wel of niet een puntje aanwezig is. 1Q t/m 3Q bepalen de evt. kleur van dat puntje.

Rond IC21 is een kleine oscillator ogebouwd voor het hardware matio laten knipoeren van een puntje.

Op de VDU-kaart zijn ook een paar wijzigingen aangebracht om de twee videosignalen (van de VDU-kaart en van het grafisch display) te mengen en de kleurinformatie toe te voegen.

Achter N34 (zie ook schema in de ELEKTUUR seot. 1983 blz. 62) komt een OR-gate (N28 en N29 die toch vrij waren). In deze OR-gate komen de beide videosignalen, de uitgang komt weer aan N31.

De kleurinformatie komt op het knoopount tussen R4 en P2.

In het schema staat bij elk IC het typenummer m.v. de soorten die volgen nu:

IC14	74S00	IC21	74S132
IC17	74LS20	IC22	74LS04
IC18	74S04	IC23	74S20
IC19	74S10	IC24	74S20
IC20	74LS00	IC25	74S86

Indien mogelijk voor IC26 t/m IC30 ook een sneller type nemen.

Aanbevolen wordt om zoveel mogelijk snelle typen te gebruiken, bv. 74S of 74ALS, maar met 74LS lukt het misschien ook wel.

DE SOFTWARE

De software zal in een volgend artikel besproken worden.
Werkend is al:

PLOT(X,Y) zet een puntje op de coördinaten X,Y.

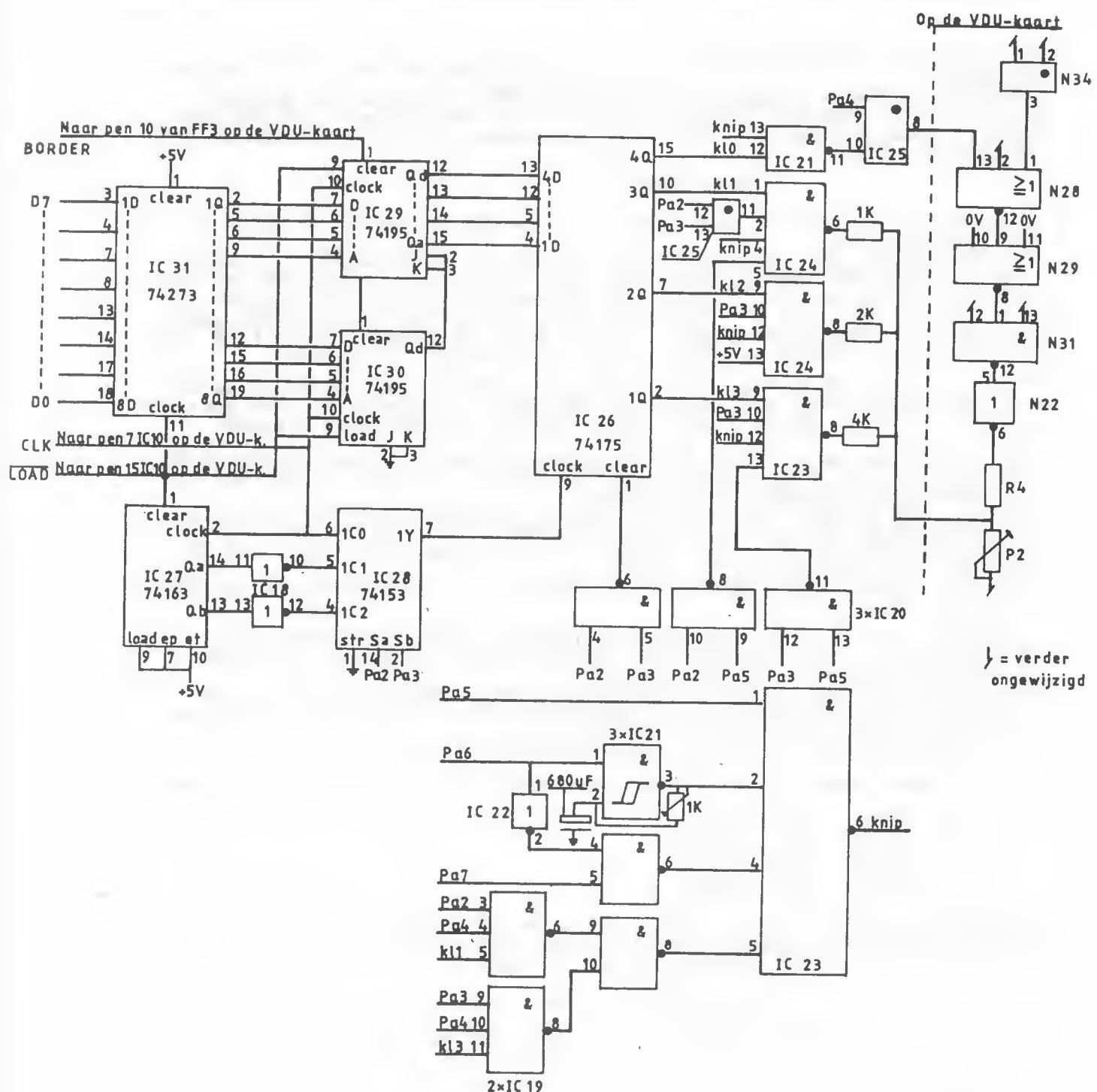
UNPLOT(X,Y) haalt het puntje weer weg.

LOOK(X,Y) kijkt of er een puntje staat op de coördinaten X,Y.

Allerlei subr. voor het werken in de kleur- en knoopmode.

Er wordt nog gewerkt aan een subr. DRAW(X1,Y1,X2,Y2) welke op een snelle manier tussen twee punten een lijn trekt.

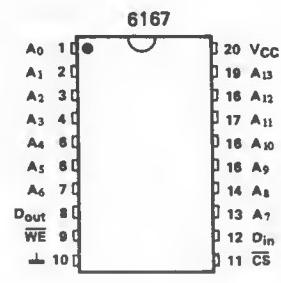
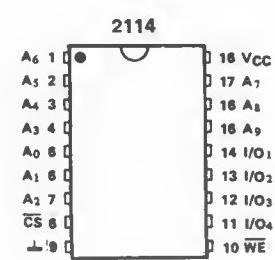
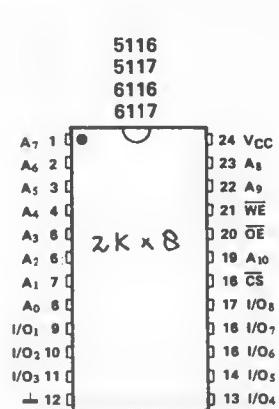
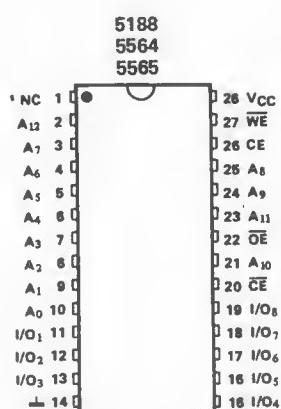
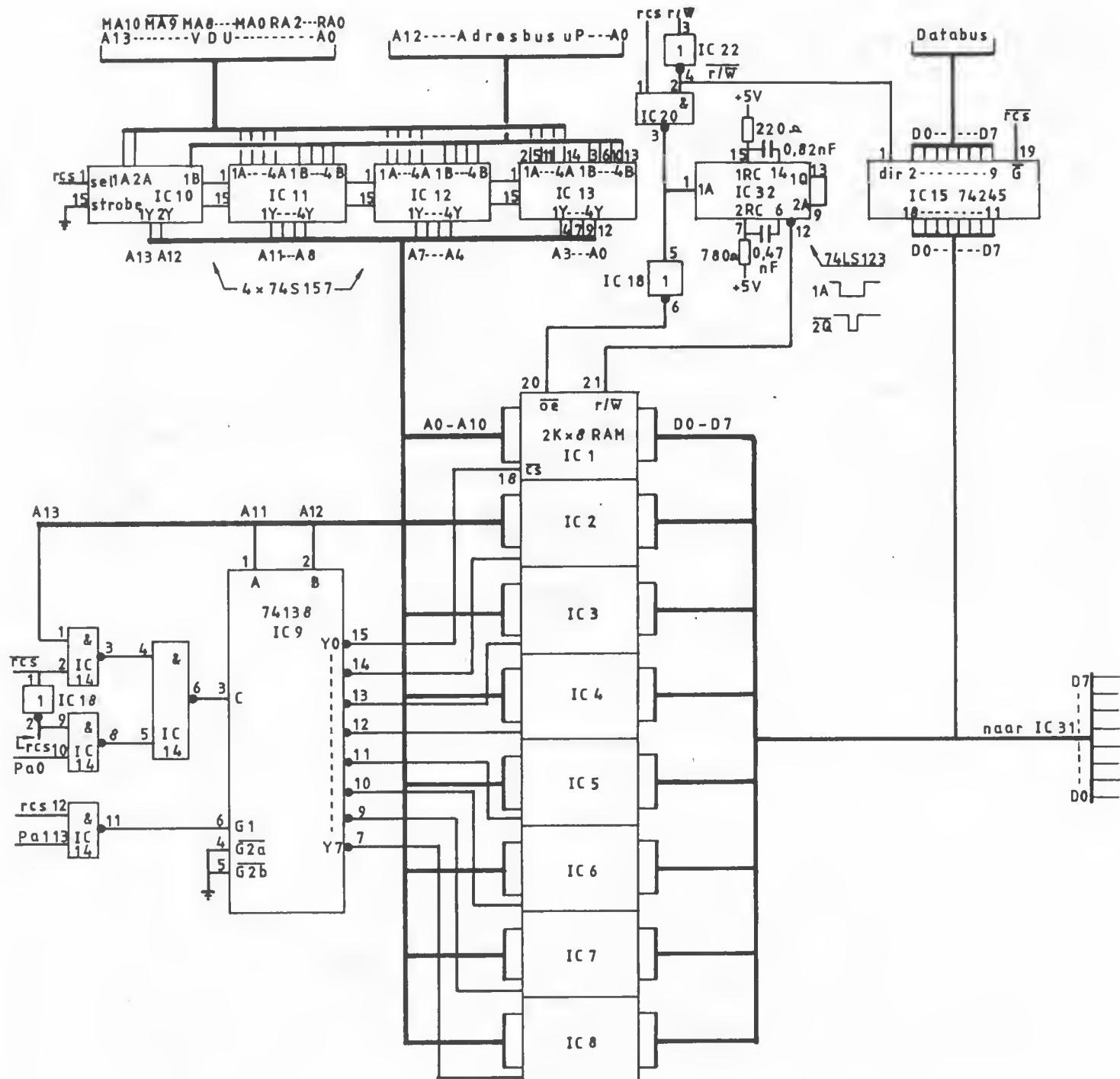
JJA en JAJ JANSSEN
Gerardsweg 30
6525 RT NIJMEGEN
Tel.: 080-562082



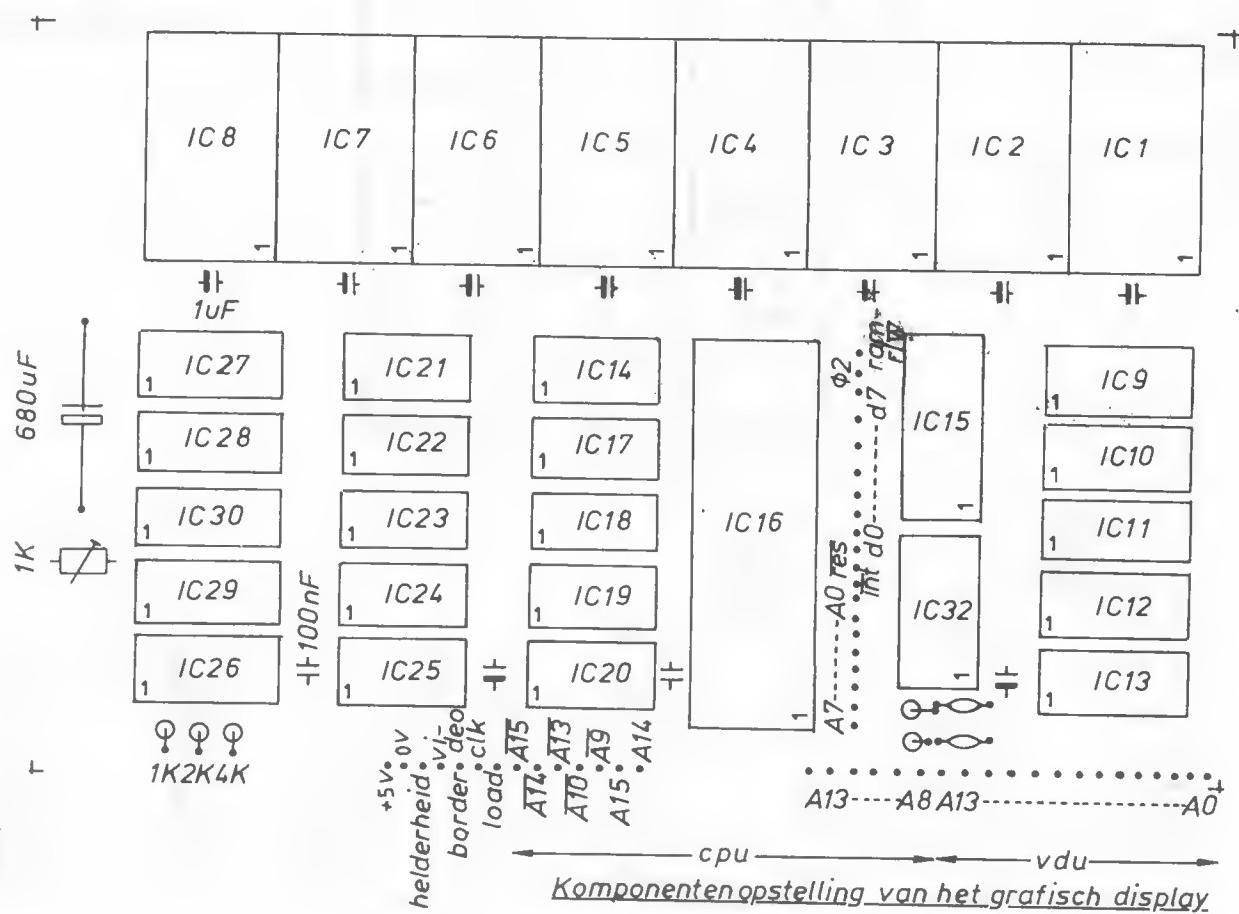
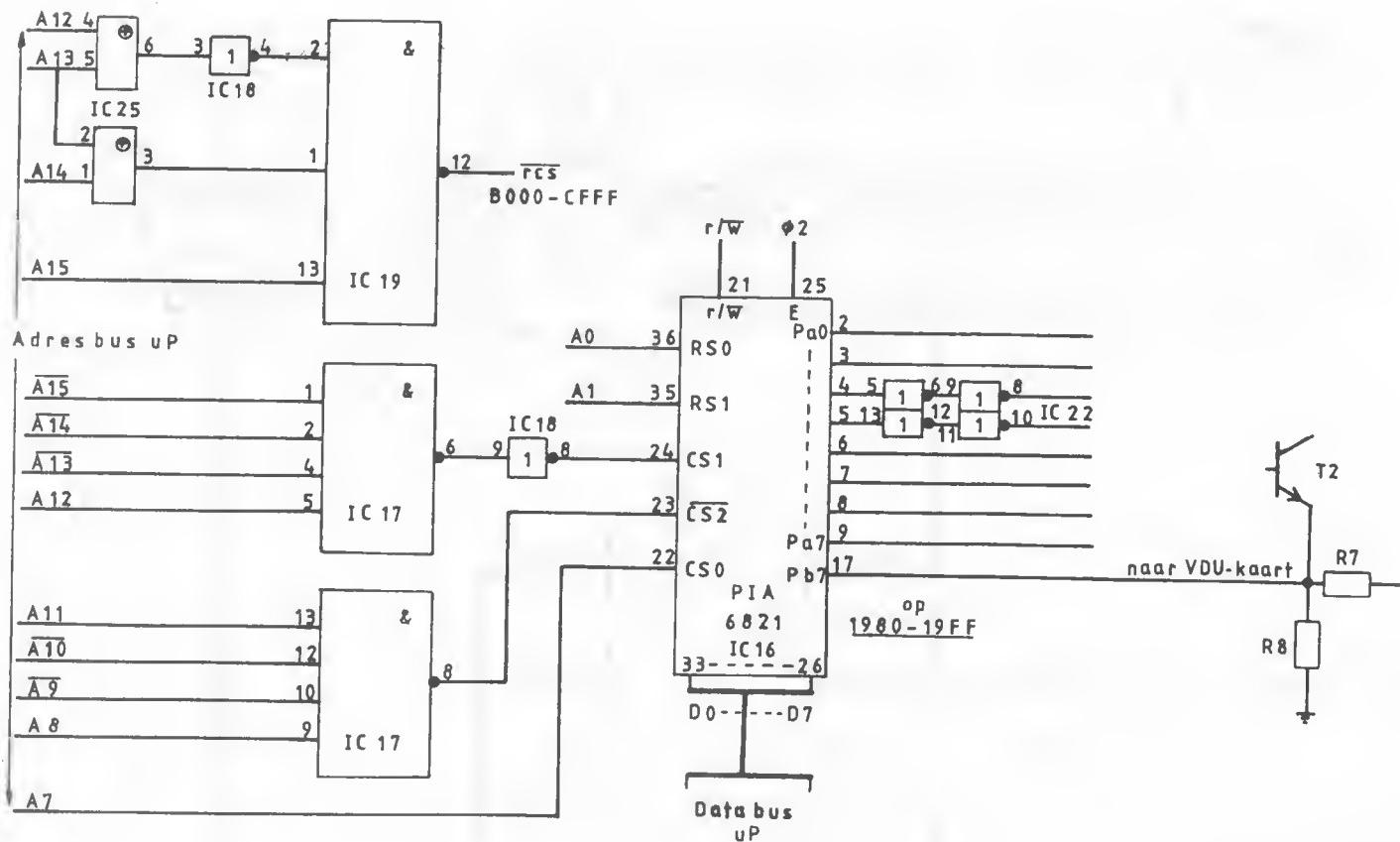
b7	b6	b5	b4	b3	b2	b1	b0	adres 0
adres 1								
adres 7								adres 15

Figuur 1
Onderbouw van het scherm.

DE 6502 KENNER



9



>PR

DE 65SC816. EEN 16 BITS OPVOLGER VOOR DE 6502

Sinds Chuck Peddle in 1975 de NMOS 6502 microprocessor bij MOS Technology ontwikkelde, is er aan deze chip maar weinig veranderd. De instructieset en het hardware-concept van de 6502 bleken zo goed doordacht te zijn dat nu -9 jaar later- de 6502 nog steeds een van de best verkochte processors is. Sinds kort realiseren de fabrikanten zich echter dat dit niet eeuwig zo zal blijven. Vooral de beperkte adresseercapaciteit en de hoge vermodendissipatie van de 6502 leveren problemen op. Systemen met meer dan 64 K RAM zijn tegenwoordig al heel gewoon, en NMOS processors zijn moeilijk in draagbare apparatuur toe te passen. Aan het laatste is inmiddels teemoet gekomen door de ontwikkeling van de 65C02 die geheel bin-compatibel is met de NMOS versie en die bovendien nog een uitbreiding van de instructieset kent.

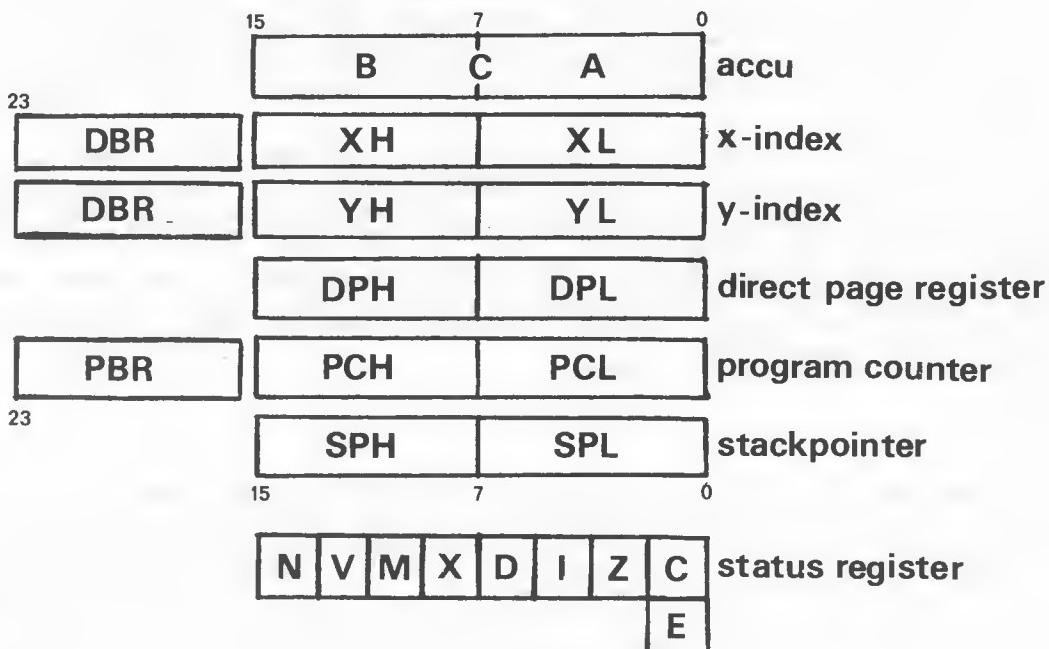
Kort geleden is een nieuwe processor op de markt gekomen die het adresseerprobleem oplost. De nieuwe processor draagt het nummer 65SC816 en kan gezien worden als de CMOS 16 bits opvolger van de 6502. De 65816 is zowel in staat zich als een 8 bits en als een 16 bits processor te gedraagen. Een enkel bit in het status register bepaalt of de processor zich als een 65C02 of als een 65816 gedraagt. Is dit E bit 1 dan wordt de 65C02 na-gebootst echter zonder de extra instructies die de firma ROCKWELL er in meende moeten te staan.

Maken we het E bit 0 dan hebben we plotseling een geheel andere processor in handen. De accumulator en de index registers zijn 16 bits breed geworden. De programcounter wordt uitgebreid met een 8 bits program bank register. Via dit bank register, dat gemultiplieert wordt met de databus, kunnen 256 geheugenbanken van 64 K worden geselecteerd. Hierdoor is de 65816 in staat om 16 Mega (!) byte geheugen te adresseren. De stackpointer is eveneens 16 bits breed en kan daardoor willekeurig in geheugenbank 0 gezet worden. Verder wordt er, naar analogie met de 6809, een direct-page register toegevoegd. Hierdoor is het mogelijk de zero page (die nu direct page heeft) eveneens willekeurig in bank 0 te plaatsen.

Het is nog steeds mogelijk de accu en de indexregisters als 8 bits eenheden te behandelen door twee bits in het statusregister te zetten of te resetten.

Het programmeermodel van 65816 verduidelijkt een en ander.

65SC816



NIEUWE ADRESSEERINGSMODES

De direct (=z-page) indirect y indexed, de direct indirect, de absolute, de absolute x indexed en de relative adresseringsmode (leest u het gerust nog maar een keer over) zijn uitgebreid met een "long" mode. De direct page instructies krijgen nu een 16 bits ooperand, waardoor elke direct page in de 256 geheugenbanken geadresseerd kan worden. De absolute instructies krijgen een 24 bits ooperand waardoor elk adres in het 16 Mbyte bereik geadresseerd kan worden. Ook de branch oordrachten kunnen voorzien worden van een 16 bits offset.

Drie nieuwe adresseringsmodi hebben te maken met stack acces: stack, stack relative en stack relative indirect indexed. Met behulp hiervan kan een subroutine op een eenvoudige manier parameters van de stack halen die er door een andere routine oopezet zijn.

NIEUWE INSTRUCTIES

Aanpeuzien de 65816 compatible is met de 65C02, zijn de 65C02 instructies aanwezig. Deze zijn al eens eerder behandeld in de 6502 KENNER en derhalve da ik er nu niet verder op in.

Allereerst zijn er nieuwe instructies die te maken hebben met de extra registers van de 65816:

PHB	push and pull data bank registers from stack
PHD	push and pull direct page register from stack
PHK	push program bank register on stack
TCD	transfer accu C to direct page register
TDC	transfer direct page register to accu C
TCS	transfer accumulator to stackpointer
TSC	transfer stackpointer to accumulator
XBA	exchange accumulators B and A
XCE	exchange X and E flag

Verder zijn er speciale returns uit "long" geadresseerde subroutines en interrupt routines.

Drie nieuwe instructies hebben te maken met stack behandeling:

PEA	Push Effective Absolute address. Deze drie bte instructies zet de 16 bits ooperand op de stack. Deze instructie wordt o.a. gebruikt om een 16 bits constante aan een subroutine door te geven.
PEI	Push Effective Indirect address. Het tweede bte van deze instructie is een offset in de direct-page. De inhoud van dat adres plus het naastliggende wordt op de stack gezet. Deze instructie wordt bijvoorbeeld gebruikt om een indirect adres op de stack te zetten.
PER	Push Effective programcounter Relative address. Deze instructie telt een 16 bits ooperand bij de programcounter op en zet het resultaat op de stack. Hiermee kan bijvoorbeeld het adres van een datablock gered worden.

Heel interessant zijn ook de blockmove instructies MVP en MVN. Het x register bevat het eerste adres van het te verplaatsen datablock, het y register het adres waarnaar decopieerd moet worden, de accumulator het aantal te verplaatsen bytes. MVP begint met het laatste byte als eerste te verplaatsen. MVN begint met het eerste byte.

Tenslotte zijn er nog een aantal instructies toegevoegd die te maken hebben met het toepassen van de 65816 in multi-processor systemen:

COP	COProcessor instructie. Is een software interrupt met een eigen vector
STP	SToP clock instructie
WAI	WAIT for interrupt instructie.

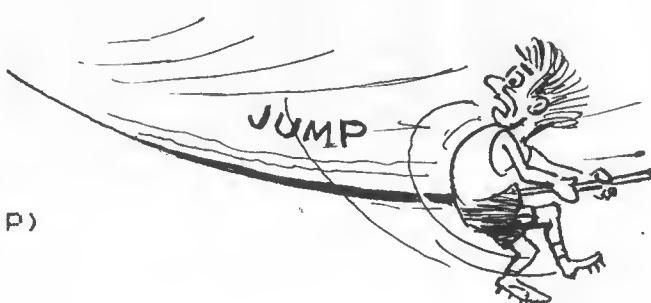
TENSLOTTE

Bovenstaande beschrijving van de 65816 is allesbehalve volledig. Hij is vooral bedoeld om een eerste indruk te geven van de mogelijkheden van de nieuwe processor zonder uitdaging in details te treden. Als meest opvallende eigenschap van de 65816 -afgezien van de 16 bits structuur- kan men vooral de nieuwe stack instructies noemen. Een stack is een flexibele, betrouwbare en eenvoudige data-structuur. Veel hogere programmeertalen (interpreters en compilers) maken gebruik van een of meerdere stacks. Kennelijk hebben de ontwerpers van de 65816 dat in gedachten gehouden toen ze zich afvroegen wat je met 16 Mbyte geheugen moet doen...

```

100 REM ****
110 REM ***** UITNODIGING TOT ..... / ****
120 REM *****> A. V d Meutter <*****
130 CALL 7037:REM CLS
140 PRINT:PRINT:PRINT:PRINT
150 PRINT" *****"
160 PRINT" **"
170 PRINT" ** U I T N O D I G I N G **"
180 PRINT" **"
190 PRINT" *****>A.V.D.MEUTTER<*****"
200 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
210 FOR Z=1 TO 5000:NEXT
230 N=100:DIM VN$(N),AN$(N)
233 FOR X=1 TO N:VN$(X)=".....":AN$(X)=".....":NEXT
234 GOTO 780
235 FOR X=1 TO N:VN$(X)=".....":AN$(X)=".....":NEXT
240 INPUT"Wie nodigt uit ":"FV$"
250 INPUT"Uw straat... ":"WO$"
260 INPUT"Uw huisnr... ":"NR$"
270 INPUT"Uw oostnr... ":"P$"
280 INPUT"Uw woonplaats. ":"WP$"
290 INPUT"Oo wat wordt uitgenodigd ? ":"DOEL$"
300 INPUT"Oo welke dag/datum ? ":"DATUM$"
310 INPUT"Om welk uur ... ":"UUR$"
320 PRINT"Geef nu de namen en voornamen(of alleen voornaam)"
330 PRINT"als er om gevraagd wordt. Na de laatste. tik \ in"
340 FOR X=1 TO N:VN$(X)=".....":AN$(X)=".....":NEXT
350 B=1:C=N
360 FOR X=B TO C
370 PRINTX:INPUT"Voornaam ... ":"VN$(X)
380 IF LEFT$(VN$(X).1)="\ THEN GOTO 420
390 INPUT"Achternaam (of MINSTENS 1 soatie) ":"AN$(X)
400 IF LEFT$(AN$(X).1)="\ THEN GOTO 420
410 NEXT
415 GOTO 440
420 PRINT"Einde invoer."
430 VN$(X)=".....":AN$(X)="....."
435 X=X-1
440 GOTO 780
450 FOR X=1 TO N
451 IF NOT LEFT$(VN$(X).1)=". " THEN NEXT X
452 IF LEFT$(VN$.1)="\ THEN VN$(X)=".....":X=X-1
453 IF LEFT$(AN$.1)="\ THEN AN$(X)=".....":X=X-1
455 INPUT"Hoeveel kaarten niet-oo-naam ? ":"EX
460 X=X+EX-1:C=X:N=X
470 REM FORMATTEER PRINTER
480 REM
490 INPUT"KAARTJES (0) OF DOORLOPEND (1) ":"C
500 PRINT"ZET PAPIER OF KAART KLAAR. EN DRUK TOETS. "
510 GETT$
520 FOR P=1 TO N
530 PRINTCHR$(17)
540 PRINTTAB(4):CHR$(27):"5":
550 PRINTCHR$(14):" UITNODIGING"
560 PRINT
570 PRINTTAB(16)
580 PRINT"OP"
590 PRINT
600 L=LEN(DOEL$)
610 TL=18-L:IF TL<0 THEN TL=0
620 PRINTTAB(TL):
630 PRINTCHR$(14):DOEL$:
640 PRINT:PRINT:PRINT"Aan : "
650 PRINTCHR$(27):"4":VN$(P):" ":AN$(P)
660 PRINT:PRINTCHR$(27):"5"
670 PRINT"Vanwege : "

```



```

680 PRINTCHR$(27) :"4":CHR$(14):FV$
690 PRINTCHR$(27) :"5"
700 PRINT"Oo ":DATUM$ :TAB(25)UUR$ :"uur."
710 PRINT
720 PRINTTWO$ :" " :NR$
730 PRINTCHR$(14):P$:CHR$(20) :" WP$"
740 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
750 PRINTCHR$(19)::IF C=1 THEN GOTO 770
760 PRINT"druk soatje om te starten":GET ST$
770 NEXT P
780 CALL7037:PRINT "WAT WIL JE ?":PRINT:PRINT
790 PRINT."* INVOER
800 PRINT."* CORRECTIE
810 PRINT."* PRINTOUT "
820 PRINT:PRINT."Kies ::::INPUT KI$
830 IF LEFT$(KI$.2)="IN" THEN N=100:GOTO235
840 IF LEFT$(KI$.2)="PR" THEN GOTO 450
850 IF LEFT$(KI$.2)="CO" THEN GOTO 880
860 PRINT"Ongeeldig.....":
870 FOR ZZ=1 TO 500:NEXT:GOTO 780
880 PRINT"WAT WIL JE CORRIGEREN ::PRINT
890 PRINT"DATUM {D} . ADRES {A} . UUR {U} . NAMEN {N}"
900 INPUT CO$
910 IF CO$="D" THEN GOTO 950
920 IF CO$="A" THEN GOTO 970
930 IF CO$="U" THEN GOTO 1000
940 IF CO$="N" THEN GOTO 1020
945 GOTO 860
950 INPUT"Datum ... ::DATUM$"
960 GOTO 780
970 INPUT"Straat... ::STA$"
980 INPUT"NR..... ::NR$"
990 GOTO 820
1000 INPUT"Uur..... ::UUR$"
1010 GOTO 820
1020 PRINT:PRINT." NAMEN":PRINT
1030 PRINT"ALLE NAMEN {A} OF EEN {E} OF DEEL {D}""
1040 PRINT"KIES. ::::INPUTKI$"
1050 IF KI$="A" THEN GOTO 340
1060 IF KI$="E" THEN INPUT"NR ? ::E:GOTO 1080
1070 GOTO 1110
1080 PRINTVN$(E):INPUT"Voornaam... ::VN$(E)"
1090 PRINTAN$(E):INPUT"Achternaam... ::AN$(E)"
1100 GOTO 820
1110 IF NOT KI$="D" THEN GOTO 820
1130 INPUT" VAN . TOT Nr ::B.C"
1140 GOTO 360
1150 REM In dit programma komen een aantal CHR$ voor welke
1160 REM de STAR DP 500 printer sturen.
1170 REM CHR$(17) of <DC-1> of Hx 11: dit is een SOFTWARE
1180 REM ON-switch (bij aanzetten niet nodig, maar wel na
1190 REM software-OFF zetten)
1200 REM CHR$(19) of <DC-3> of Hx 13: dit is een SOFTWARE
1210 REM OFF-switch. De printer geeft ACKN bij overdracht.
1220 REM zodat het systeem niet vastloopt op output, maar
1230 REM er wordt NIETS in de printer-buffer opgenomen zo-
1240 REM lang er een DC-1 code oeweeest is.
1250 REM CHR$(27) is <ESC>
1260 REM + "5" is rechte letters
1270 REM + "4" is schuine letters
1280 REM CHR$(14) of <SO> of Hx OE: zet dubbel-breed aan, en
1290 REM dat tot, ofwel een <DC-4>=CHR$(20)=Hx 14 gezonden
1300 REM wordt, ofwel een <CR> tereenkomen wordt.

```

DE 6502 KENNER

MONITOR PROGRAMMA's voor COMMODORE Computers.

Gemaakt door: Nico de Vries
Mari Andriessenrade 49
2907 MA Capelle aan den IJssel
010-502239

Inleiding.

Bij de 6502 Kenners is het eigenlijk heel gewoon dat een computer is uitgerust met een monitorprogramma. Bij de meest populaire computer, de Junior, is dit zelfs het enige vaste programma, dat dan dus tevens het operating system vormt. Ook bij andere single board microcomputers kennen we deze situatie: de KIM, de SYM en de AIM-65 zijn er voorbeelden van.

Er komt echter veelvuldig een tweede soort computers voor, namelijk computers met een BASIC-interpreter (vrijwel altijd die van Microsoft) in ROM. Deze apparaten hebben meestal geen monitor programma, en zijn daarom op machinenevaau wat moeilijk te benaderen.

Van deze laatste categorie maken ook de producten van de thans zeer succesvolle fabrikant Commodore uit.

Apparaatbeschrijvingen.

Voor Commodore begon het computertijdperk met de welbekende PET-computer. Dit apparaat met ingebouwde cassetterecoorder, beeldscherm en klein toetsenbord is vrij bekend geworden. Karakteristiek waren de afwezigheid van een monitorprogramma in de ROMs en de grote hoeveelheid bugs in zowel BASIC als het operating system.

Na een drietal jaren werd de PET vernieuwd tot de CBM-3000 serie. Deze apparaten hadden geheel nieuwe geheugenindeling in de eerste vier pagina's en een vrijwel bug-vrije ROM-inhoud, waarbij zelfs een primitief monitorprogramma niet vergeten is.

Spoedig na het verschijnen van de 3000-machines kwam de 80-koloms CBM 8032 uit, met een uitgebreidere BASIC, die voor het eerst een versienummer laat zien: BASIC 4.0. Ook in dit apparaat zat een licht verbeterd monitorprogramma.

De Commodores werden dus steeds mooier (en duurder), totdat de VIC-20 verscheen. Een speelgoed-achtig apparaat met slechts 22 tekens per regel. Dit is tevens de eerste Commodore zonder beeldscherm. Een tweede CBM primeur was een gekleurd beeld. De VIC-20 heeft, u raadt het al, geen monitor programma.

Tenslotte verscheen het kassucces de CBM-64, zo genoemd vanwege de hoeveelheid RAM in deze machine. De hardware van dit apparaat is ontegenzeggelijk fraai, de firmware is daarentegen nogal gammel; een monitorprogramma ontbreekt, terwijl een oudere versie van Microsofts BASIC is gebruikt.

TIM en MLM.

De ontwikkeling van de monitorprogramma's voor de diverse CBM machines ging analoog. Bij veel PET computers werd een bandje

meegeleverd met daarop TIM. Dit was een gewijzigde uitgave van de Terminal Interface Monitor uit de KIM, aangepast aan de in de PET aanwezige schermeditor. TIM kent zes 1-letter commando's:

.M XXXX,YYYY	Dump het geheugen van XXXX t/m YYY in hex. Veranderen met de scherm editor is toegestaan.
.R	Toon de geredde registers in hex. Ook hier is verandering met de scherm editor mogelijk.
.X	Ga terug naar BASIC.
.L XX,NAAM	Laadt het programma NAAM van device no. XX.
.S XX,NAAM,YYYY,ZZZZ	Save het blok geheugen van YYY tot ZZZZ op device no. XX.
.G XXXX	Voer het programma op adres XXXX uit. Hierbij worden vooraf de CPU-registers met de waarden van het R-commando geladen. Indien XXXX niet wordt opgegeven, wordt de geredde PC gebruikt.

Ik geef de commando's omdat er slechts zelden een handleiding bij de computer werd geleverd; niet voor niets is de oude PET de slechtst gedocumenteerde computer aller tijden.

TIM wordt op dezelfde plaats in geheugen geladen als een BASIC programma: dit heeft verscheidene nadelen, waaronder de onmogelijkheid de monitor en een BASIC programma tegelijkertijd in het geheugen te hebben.

Dit nadeel werd opgeheven in de 3000-serie: deze machines hebben een verbeterde TIM in de ROMs ingebouwd. Ook wijst bij deze apparaten de BRK-vector naar de monitor, die MLM heet (staat simpelweg voor: Machine Language Monitor). Dit gegeven is er de oorzaak van, dat in PET/CBM kringen de monitor vrijwel altijd gestart wordt via een BRK (met SYS4 of SYS1024), ondanks het feit dat een keurige warme start is ingebouwd: SYS64785.

MLM heeft twee belangrijke verbeteringen ten opzichte van TIM. Ten eerste hebben de L en S commando's een andere syntax, waardoor defaults mogelijk werden. De tweede verbetering wordt gevormd door een softvector in RAM, waardoor er commando's aan de monitor kunnen worden toegevoegd. Tegenover de verbeteringen staan een paar typische bugs die zijn overge-erfd van TIM, het accepteren van andere letters dan A t/m F in hex getallen is hier een voorbeeld van. De MLM heeft dezelfde commando's als TIM.

De MLM in BASIC 4.0 is op 1 punt gewijzigd: bij aanroep via een BRK wordt de uitvoer teruggezet naar het scherm. Tot zover de standaard monitoren voor de Commodores.

SUPERMON.

In het voorgaande werd uitgelegd dat de MLM in de 3000-serie en in BASIC 4.0 kan worden uitgebreid met extra commando's. Het oudste, en tevens meest bekende programma dat dit doet is SUPERMON. Deze supermonitor (nou ja...) werd gemaakt door Jim Butterfield (een der grootste PET/CBM guru's) en Bill Seiler, toenertijd een werknemer van CBM. SUPERMON verscheen in verplaatsbare vorm, compleet met een relocater in BASIC, die het programma opbergt in de top van het RAM, waar die top dan ook mag zijn. Later versies van SUPERMON (die voor BASIC 4.0 en de 64) hebben een relocater in machinetaal.

SUPERMON voegt in ieder geval de volgende commando's toe:
.F XXXX,YYYY,ZZ Vul het geheugen van XXXX t/m YYY met

- .T XXXX,YYYY,2222 het byte 22.
 Verplaats het blok geheugen XXXX t/m YYYY
 naar adres 2222.
- .D XXXX Disassembleer 22 instructies, te beginnen
 op adres XXXX. De getoonde hexadecimale
 bytes mogen met de schermeditor veranderd
 worden.
- .A XXXX instructie Eenvoudige assembler. De assembler
 accepteert mnemonics met hexadecimale
 operanden. Een relatief adres bij een
 branch mag absoluut gegeven worden; de
 offset wordt door de assembler
 uitgerekend. Na de vertaling verschijnt op
 de volgende regel automatisch een A en het
 volgende instructie adres, zodat u alleen
 de instructie behoeft in te typen; leve de
 schermeditor!
- .H XXXX,YYYY,22,AA Zoek het geheugenblok XXXX t/m YYYY af
 naar de byte-combinatie 22AA. Het aantal
 bytes mag maximaal 32 zijn. De adressen
 van de gevonden plaatsen worden op het
 scherm afdrukt.
- .H XXXX,YYYY,'STRING' Zoek het geheugenblok XXXX t/m YYYY af
 naar de string STRING. De string mag
 maximaal 32 tekens lang zijn. De adressen
 van de gevonden plaatsen worden op het
 scherm afdrukt.
- .I Single step. Naar keuze wordt het
 programma dat begint op het adres van de
 geredde PC stap voor stap, danwel langzaam
 uitgevoerd. Hierbij wordt iedere
 uitgevoerde instructie gedisassembleerd en
 worden de waarden van de registers
 getoond. (Niet in de CBM-64).
- Er zijn in de loop der tijd een aantal varianten op SUPERMON
verschenen die 1 of twee commando's meer of minder hebben.
Afhankelijk van de versie kunnen ook nog aanwezig zijn:
- .C XXXX,YYYY Bereken offset. De branch staat op XXXX en
 moet wijzen naar YYYY.
- .P XXXX,YYYY Disassembleer het blok geheugen van XXXX
 t/m YYYY. Speciaal bedoeld voor de printer
 die vooraf geactiveerd moet worden.
- De SUPERMON voor de CBM-64 omvat naast de nieuwe commando's
ook de vroegere MLM commando's omdat dit apparaat geen MLM in
de ROMs heeft.
- De SUPERMON-versies voor de 3000-serie, BASIC 4.0 en de CBM-
64 zijn alle gemaakt door Jim Butterfield en bevinden zich in
het Public Domain. Dit houdt in, dat deze programma's vrij
gekopieerd mogen worden; er rusten GEEN auteursrechten op.
- Naast de Public Domain versies bestaan er ook verscheidene
versies voor de oude PET (met ingebouwde TIM) en de VIC-20,
meestal door een fanaat via een listing omgeschreven voor
deze apparaten.

EXTRAMON.

Voor sommigen was SUPERMON niet mooi en uitgebreid genoeg.
Dus maakte Bill Seiler een grotere monitor, EXTRAMON geheten.

EXTRAMON kan alles wat SUPERMON kan plus (uiteindelijk) nog meer. De extra commando's zijn:

- .B XXXX,YY Zet een breekpunt op XXXX. Dit breekpunt moet YY maal gepasseerd worden voordat op single step wordt overgegaan.
- .Q XXXX Voer het programma op XXXX uit totdat het met B ingestelde breekpunt is bereikt en ga vervolgens over op single step.
- .E Zet de noodstop aan. Legt de interrupt om naar een STOP-toets test zodat een programma met deze toets onderbroken kan worden.
- .U Zet de noodstop weer uit.
- .I XXXX,YYYY Dump het geheugenblok XXXX t/m YYYY in hex en ASCII. Veranderen met de schermeditor is toegestaan. (I uit SUPERMON vervalt: zie Q).
- .N XXXX,YYYY,ZZZZ,AAAA,BBBB Een relocator die zo ingewikkeld is dat niemand hem ooit gebruikte: Verschuif het blok XXXX t/m YYYY naar ZZZZ en verander alle absolute adressen in het gebied AAAA t/m BBBB die op het verplaatste stuk betrekking hebben. De tweede stap is dan:
.N XXXX,YYYY,ZZZZ,AAAA,BBBB W Pas vervolgens adrestabellen aan, met dezelfde criteria als boven.

EXTRAMON is er voor de 3000-serie en BASIC 4.0. Het is nooit erg populair geworden in Nederland. Versies voor de PET, de VIC-20 en de CBM-64 ontbreken.

MICROMON.

Onafhankelijk van het door Commodore zelf verspreide EXTRAMON ontstond MICROMON, de meest uitgebreide monitor voor CBM-computers. MICROMON heeft als bijzonderheid dat het voor- en ACHTERwaards kan dumpen en disassembleren, simpelweg door respectievelijk de cursor-down dan wel de cursor-up toets vast te houden. Hierbij zorgt MICROMON voor de repeat op de toets. Karakteristiek voor MICROMON is, dat het geen gebruik maakt van de in het ROM aanwezige MLM. De MLM is hiertoe in MICROMON zelf opgenomen. Dit is gedaan om het dump via de cursor-toetsen mogelijk te maken, en de ASCII-dump aan het M-commando toe te voegen. Verder is MICROMON een combinatie van SUPERMON en EXTRAMON.

De volgende commando werden naast de reeds bestaande toegevoegd:

- .B Zet breekpunt, zie B onder EXTRAMON.
- .C XXXX,YYYY,ZZZZ Vergelijk het geheugen van XXXX t/m YYYY met het blok dat begint bij ZZZZ.
- .E Ga naar BASIC en schakel de MICROMON interrupt uit.
- .K Ga naar MLM en schakel de MICROMON interrupt uit. Dit is nodig om in de MLM op cassette te kunnen saven.
- .Q Single step, zie Q onder EXTRAMON.
- .O XXXX,YYYY Bereken offset, zie C onder SUPERMON.
- .W Single step, zie I onder SUPERMON.

DE 6502 KENNER

.Z	Toggle karakterset.
.\$XXXX	Hex naar decimaal, ASCII en binair conversie.
.#XXXXX	Decimaal naar hex, ASCII en binair conversie.
.%XXXXXXXXXXXXXXXXX	Binair naar hex, ASCII en decimaal conversie.
. "string	ASCII naar hex, decimaal en binair conversie.
.+ XXXX,YYYYY	Tel XXXX en YYYYY bij elkaar op.
.- XXXX,YYYYY	Trek XXXX en YYYYY van elkaar af.
.& XXXX,YYYYY	Bereken de checksum van het blok XXXX t/m YYYYY.

MICROMON kan indien gewenst worden uitgebreid, en heet dan: MICROMON Plus. De extra commando's zijn:

.d XXXX,YYYYY	Disassembler voor de printer. Zie P bij SUPERMON.
.I "NAAM"	Laadt het programma NAAM van schijf die default is i.p.v. cassette 1.
.m XXXX YYYYY	Dump XXXX t/m YYYYY in hex en ASCII, 16 bytes per regel. Voor de printer.
.P	Schakel de printer aan of uit.
.I "HEADER"	Schakel een kopregel voor de printer in.
.J "NAAM"	Laat het laadadres van programma NAAM zien. Alleen voor schijf.
.Y XXXX,"NAAM"	Laadt het programma NAAM op adres XXXX. Alleen voor schijf.

Naast deze commando's is ook nog een DOS-hulp opgenomen, die bij CBM-schijfgebruikers bekend is als DOS-support.

MICROMON (Plus) is er voor de 3000-serie en BASIC 4.0. Een versie voor de CBM-64 bestaat ook, maar heeft niet alle commando's van de versies voor de andere apparaten.

NEWTIM S.

Een monitor die geheel los staat van de vorige monitoren. NEWTIM S is van onduidelijke afkomst: de geruchten wijzen in de richting van Zwitserland. NEWTIM S en EXTRAMON zijn ongeveer even krachtig, waarbij NEWTIM S een betere ondersteuning van de schijfunit geeft, en een mooiere implementatie van de printer heeft ingebouwd. Vanwege de (mogelijk) Zwitserse afkomst moeten de commandoletters met Duitse i.p.v. Engelse woorden ge-associeerd worden. De extra commando's naast die van MLM zijn:

.A XXXX instr.	Assembler, zie A bij SUPERMON.
.B XXXX,YY	Zet Breekpunt, zie B bij EXTRAMON.
.C XXXX,YYYY,2222	Verschuif een blok geheugen. Zie T bij SUPERMON.
.CU XXXX,YYYY,2222	Reloceer een blok geheugen.
.D XXXX	Disassembler, zie D bij SUPERMON.
.D XXXX,YYYY	Disassembler, zie P bij SUPERMON.
.D XXXX-	Disassembler van XXXX t/m \$FFFF.
.E XXXX	Execute tot breekpunt. Zie Q bij EXTRAMON.
.F XXXX,YYYY,22	Vul geheugen, zie F bij SUPERMON.
.H XXXX,YYYY,/22/	Zoek naar byte(s).
.H XXXX,YYYY,/22/AA/	Als vorige, doch vervang 22 door AA.
.H XXXX,YYYY'/string/	Zoek naar string.
.H XXXX,YYYY'/string1/string2/	Als vorige, vervang string1 door string2.

.I XXXX,YYYY	Dump in hex en ASCII, zie I bij EXTRAMON.
.J XXXX-	Dump in hex en ASCII, van XXXX t/m \$FFFF.
.K XXXX,YYYY	Controleer het blok XXXX t/m YYYY op illegale opcodes.
.L XXXX-	Als vorige, doch eindadres=\$FFFF.
.O XXXX	Single step, huidige instructie in de top-regel van het scherm, zodat de uitvoer van het uitgevoerde programma zichtbaar blijft.
.TE	Schakelt de printer in. De uitvoer van de commando's D, E, H, I, K, en W verschijnt nu ook op papier, met automatische paginering.
.TA	Schakelt de printer uit.
.TT tekst	Stuur de string tekst naar de printer.
.V XXXX,YYYY,ZZZZ	Zie C bij MICROMON.
.W XXXX	Single step, zie I bij SUPERMON.
.Y	Toont de CPU-vlaggen in binair. Ze mogen met de editor binair veranderd worden.
.Z	Zet de interrupt op de default.
.\$ XXXX	Converteer van hex naar decimaal.
.# XXXXX	Converteer van decimaal naar hex.
.% XXXY	Converteer van hex naar binair.
.&+ XXXX,YYYY	Tel XXXX en YYYY bij elkaar op.
.&- XXXX,YYYY	Trek XXXX en YYYY van elkaar af.
.&A XXXX,YYYY	Bereken XXXX AND YYYY.
.&O XXXX,YYYY	Bereken XXXX OR YYYY.
.&E XXXX,YYYY	Bereken XXXX EOR YYYY.
.&E XXXX-	Inverteer XXXX.
Bijzonder aan NEWTIM S is het feit dat het H-commando niet alleen kan zoeken, maar ook nog vervangen. Dit commando wordt nog fraaier door de mogelijkheid met don't cares te zoeken, EN en te vervangen. Tenslotte bevat NEWTIM S de DOS-support, die niet alleen in de monitor, maar ook in BASIC werkt. NEWTIM S is er voor de oude PET, de 3000-serie, en alle versies van BASIC 4.0. Voor de 64 moet hij nog worden aangepast.	
Als NEWTIM C bestaat een versie aangepast aan de GTE/Synertek 6502 CPU.	
De auteur van dit verhaal gebruikt NEWTIM C.	

Besluit.

Zo, dat was het dan. Een overzicht van alle monitorprogramma's voor de meeste Commodore computers. Voor de 3000-serie is iedere monitor beschikbaar; voor de VIC vrijwel geeneen.

De 64 komt er ook wat bekaaid af: SUPERMON en MICROMON.

DE VOLGENDE EDITIES VAN DE 6502 KENNER ZIJN NOG VERKRIJGBAAR DOOR HET STORTEN OF OVERSCHRIJVEN VAN FL. 9.00 PER EXEMPLAAR OP GIROREKENING 3757649 TEN NAME VAN DE PENNINGMEESTER KIM GEBRUIKERSCLUB NED. TE KRIMPEN A.D. IJSEL. ONDER VERMELDING VAN DE GEWENSTE EDITIES.

NR. 16	MEI 1981	NR. 17	AUG 1981
NR. 18	OKT 1981	NR. 19	DEC 1981
NR. 20	NIET MEER LEVERBAAR	NR. 21	NIET MEER LEVERBAAR
NR. 22	AUG 1982	NR. 23	OKT 1982
NR. 24	DEC 1982	NR. 25	FEB 1983
NR. 26	MEI 1983	NR. 27	AUG 1983
NR. 28	OKT 1983	NR. 29	DEC 1983
NR. 30	FEB 1984	NR. 31	APR 1984
NR. 32	JUN 1984	NR. 33	AUG 1984
NR. 34	OKT 1984	NR. 35	DEC 1984

FORTH OP JUNIOR COMPUTERS DEEL 1.

door: G. van Opbroek
Hooglanden 28
9801 LB Zuidhorn

(GEVOP)

1. Inleiding.

Het is mijn bedoeling het gebruik van de taal FORTH binnen de KIM-club te stimuleren en te proberen enige duidelijkheid te verschaffen in de verschillen die bestaan tussen de verschillende versies van FORTH die door de leden gebruikt worden. Ik ben van plan in een serie artikeltjes een aantal zaken aan bod te laten komen die voor de beginnende en ervaren FORTH programmeur interessant zijn. Ik denk hierbij onder andere aan de volgende zaken:

- Een algemene inleiding over FORTH.
- Standaards (fig-FORTH, 79-standaard, 83-standaard, PE-FORTH voor SENIOR).
- Het gebruik van de EOITOR.
- Het gebruik van de ASSEMBLER.
- TAPE-ROUTINES voor onderlinge uitwisseling.
-

Reacties op (en bijdragen voor) deze serie publicaties kunt u kwijt bij de auteur van dit artikeltje en de redactiesecretaris. Om praktische redenen verzoeken wij u uitsluitend schriftelijk te reageren.

Om mijzelf enigzins te plaatsen lijkt het mij goed mijzelf een beetje in te leiden. Ik ben in het bezit van een PROTON SENIOR systeem en werk hierop vrijwel uitsluitend in FORTH en ASSEMBLER. Ik werk al vanaf het moment dat mijn systeem operationeel was (januari '84) in FORTH. In eerste instantie was dit de door de firma PROTON geleverde PE-FORTH. Daar deze versie echter zeer sterk van de min of meer algemene standaard FORTH afwijkt ben ik ertoe over gegaan mijn ontwikkelde programma's ook geschikt te maken voor de FORTH die via de cassettebibliotheek verspreid wordt. Ik heb daarom deze FORTH aangeschaft, compleet met de volledige set documentatie. Omdat deze FORTH niet met disk kan samenwerken heb ik vanuit de sourcelisting de fig-FORTH opnieuw geïnstalleerd. Ik heb deze FORTH geschikt gemaakt voor het werken met schijven onder het PROTON DOS en uitgebreid met de EOITOR, een zelf ontwikkelde ASSEMBLER en de fig-79 standaard conversies. Er wordt aan gewerkt om deze versie van FORTH geschikt te maken voor meerdere configuraties rond een JUNIOR en dan te verspreiden via de cassettebibliotheek.

2. Algemene inleiding over FORTH.

FORTH is ontwikkeld door Charles H. More om de telescopen van het Kitts Peak Observatorium te besturen. De reden van het ontwikkelen van een nieuwe taal was het feit dat eigenlijk niet een van de toen bestaande talen goed geschikt was voor besturingsdoeleinden. Ik ben van mening dat More er inderdaad zeer goed in geslaagd is een taal te ontwikkelen voor besturingen. Aan de hand van een voorbeeld zal ik trachten dat duidelijk te maken:

Stel we laten een modelspoorbaan door onze JUNIOR besturen en gebruiken hiervoor de taal FORTH. We hebben de wissels genummerd van 1 t/m 10 en de seinen van 1 t/m 20. Voor het station staat een trein te wachten om op spoor 1 binnen te komen. We hebben stukjes FORTH geschreven om wissels om te zetten (WISSEL-RECHT,

SEIN-GROEN) en om seinen om te zetten (SEIN-ROOD, SEIN-GROEN). Om de trein binnen te laten komen kunnen we dan intypen:

I WISSEL-KROM 2 WISSEL-RECHT 5 SEIN-ROOD 1 SEIN-GROEN en de trein zal binnenkomen en wachten voor sein 5.

We kunnen zelfs een stap verder gaan door het volgende te doen:

: >SPOOR1 I WISSEL-KROM 2 WISSEL-RECHT 5 SEIN-ROOD 1 SEIN-GROEN ;

We hebben nu de definitie >SPOOR1 gemaakt en de trein zal op spoor 1 binnenkomen als we intypen:

>SPOOR1

Hoewel het bovenstaande voorbeeld misschien de illusie geeft dat FORTH alleen geschikt is voor besturingen is dat niet juist. Ik ben van mening dat FORTH een moderne gestructureerde taal is die voor bijna alle toepassingen geschikt is. Het enige waar FORTH minder geschikt voor is is het uitvoeren van ingewikkelde berekeningen omdat floating-point getallen ontbreken. Er wordt door mij gewerkt aan een pakket voor het werken met floating-point getallen in FORTH. Ik hoop hier in deze serie artikeltjes nog op terug te komen.

Afgezien van floating-point heeft FORTH slechts een nadeel. FORTH is een taal die sterk afwijkt van alle andere talen. Het gevolg hiervan is dat het voor een beginner moeilijk is zich FORTH eigen te maken en programma's te schrijven die echt FORTH-stijl zijn. Dit wordt voornamelijk veroorzaakt door het feit dat FORTH gebruik maakt van een zogenaamde "STACK". FORTH kent wel variabelen, maar deze moeten meer opgevat worden als geheugen zoals bijvoorbeeld de geheugens op zakrekenmachines gebruikt worden. Het gevolg van dit alles is dat bewerkingen in de zgn. Reversed Polish Notation (RPN) geschreven moeten worden. D.w.z. het optellen van 12 en 23 wordt geschreven als:

12 23 +

Deze notatie is exact gelijk aan de notatie die bij de HP zakrekenmachines gebruikt wordt.

Een ander kenmerk van FORTH is dat alles dat gedefinieerd wordt onderdeel wordt van de taal zelf. Dat betekent dat alles wat gedefinieerd is gebruikt kan worden in nieuwe definities maar ook vanaf het toetsenbord uitgevoerd kan worden. Dit maakt dat het testen van nieuwe definities (in FORTH heten dit "woorden") heel eenvoudig wordt. Uit het bovenstaande voorbeeld kunnen eerst de afzonderlijke woorden WISSEL-RECHT en SEIN-ROOD getest worden waarna >SPOOR1 getest kan worden. Verder is het zo dat er bijvoorbeeld een nieuw woord TREINLOOP gemaakt kan worden waar >SPOOR1 weer een onderdeel van vormt.

Intern bestaat FORTH uit een woordenlijst (DICTIONARY) waarin alle gedefinieerde woorden staan, gevolgd door verwijzingen naar de woorden waaruit de definitie bestaat. Deze verwijzingen worden tijdens het laden van het FORTH woord ingevuld. Als FORTH vanaf het toetsenbord een opdracht krijgt zoekt hij in de woordenlijst de opdracht (het woord) op en voert de woorden waaruit het gevonden woord bestaat uit. De verwijzingen naar de andere woorden zijn gecompileerd zodat FORTH niet opnieuw naar deze woorden hoeft te zoeken. Om deze reden wordt FORTH een interpreter/compiler genoemd. (De opdrachten worden geïntepreteerd maar de verwijzingen zijn gecompileerd).

3. Gebruik van de stack.

Hoewel ik absoluut niet van plan ben een cursus FORTH te schrijven zijn er toch enkele zaken waar ik aandacht aan wil besteden. Voor boeken waaruit FORTH geleerd kan worden verwijst ik naar het hoofdstuk literatuur.

Zoals al is opgemerkt, wijkt het werken met getallen in FORTH sterk af van wat wij over het algemeen gewend zijn. Dit betekent dat beginners er vrij veel moeite mee hebben programma's te schrijven waarbij echt in FORTH-stijl geprogrammeerd wordt. Ik wil in dit hoofdstuk enkele handreikingen geven voor het in stijl werken met de stack. In de gegeven voorbeelden zal ik gebruik maken van de notatie die ook door de FORTH INTEREST GROUP gebruikt wordt. Deze notatie heeft de vorm:

(n1 n2 - n3 n4)

waarbij de getallen links van "-" voor het uitvoeren van het betreffende woord op de stack staan en de getallen rechts van "-" na het uitvoeren van het woord.

Een getal wordt op de stack gezet door dit getal in te typen. Het verwijderen van een getal van de stack gebeurt door het getal af te drukken (met behulp van het FORTH woord ".") of door het FORTH woord "DROP".

Alle berekeningen vinden plaats met getallen die op de stack staan. Het resultaat van deze berekeningen wordt ook weer op de stack geplaatst:

Bijvoorbeeld het FORTH woord "+" :

(n1 n2 - n3)

telt de getallen n1 en n2 op de stack op en plaatst de som n3 weer als resultaat op de stack. Hetzelfde geldt voor alle andere rekenkundige bewerkingen. IK beperk me bij deze voorbeelden uitsluitend tot gehele zgn. "single precision" getallen in de range -32768 t/m +32767. Het is mogelijk met zgn. "double precision" getallen te werken die een veel groter bereik hebben. In PE-FORTH is het zelfs mogelijk met floating-point getallen te werken. IK wil hier op dit moment niet op ingaan. Voor details over deze zaken verwijst ik naar de literatuur en de handleiding van PE-FORTH.

Ten opzichte van de normale (infix) notatie heeft de RPN (postfix) notatie het voordeel dat het gebruik van haakjes geheel overbodig is.

Voorbeeld: $(2+3) \times (5+15) / ((4+1) \times (7-3))$ wordt:
2 3 + 5 15 + X 4 1 + 7 3 - X /

waarna het resultaat op de stack komt te staan.

Zoals al is opgemerkt kan er in FORTH met variabelen gewerkt worden. Voor het begrijpen van de volgende voorbeelden is het noodzakelijk even te laten zien hoe dat gaat.

- 1) Definitie van een variabele: VARIABLE naam
- 2) Een variabele een waarde geven: waarde naam !
- 3) Een waarde uit een variabele op de stack zetten: naam @

Opmerking: Deze voorbeelden zijn in 79-standaard FORTH, in andere versies van FORTH kunnen afwijkingen zitten. IK kom daar in een later stadium nog op terug.

Het lijkt logisch variabelen te gebruiken om waarden aan woorden door te geven zoals bijvoorbeeld in BASIC of COMAL subroutines gebruikelijk is. Een FORTH programma om een tafel van vermenigvuldiging af te drukken ziet er dan als volgt uit:

```
VARIABLE MULCNT      VARIABLE MULCTR
VARIABLE PRODUCT

: TAFEL 21 I DO I MULCTR ! MULCNT @
    MULCTR @ X PRODUCT !
    CR MULCTR @ ." X"
    MULCNT @ . ." = " PRODUCT @ .
LOOP
;
```

Aanroep bijvoorbeeld: 8 MULCNT ! TAFEL

Als we echter alle berekeningen op de stack uitvoeren krijgen we een programma in FORTH-stijl zonder het gebruik van variabelen:

```
: TAFEL ( n - ) ( drukt voor n de tafel van verme- )
    ( nigvuldiging af
    21 I DO CR I DUP . ." X" OVER DUP .
        ." = " X .
    LOOP DROP
;
```

Aanroep bijvoorbeeld: 8 TAFEL

Om de zaken nog iets duidelijker te maken geef ik ook het equivalent van het programma in BASIC:

```
10 INPUT MULCNT
20 GOSUB 1000 : REM TAFEL
30 END
1000 FOR MULCTR=1 TO 20
1010 PRINT : PRODUKT=MULCTR*MULCNT
1020 PRINT MULCTR;"X";MULCNT;"=";PRODUKT;
1030 NEXT MULCTR
1040 RETURN
```

IK ben van mening dat programma's in FORTH-stijl zo min mogelijk gebruik moeten maken van variabelen. Dit betekent dat het doorgeven van parameters zoveel mogelijk via de stack moet gebeuren en dat tussenresultaten ook op de stack moeten worden bewaard. Deze aanpak heeft wel als nadeel dat de leesbaarheid van FORTH programma's erg slecht kan worden. Het is daarom erg belangrijk het programma rijkelijk van commentaar te voorzien van wat en in welke volgorde er op de stack staat. IK vind dat variabelen gebruikt moeten worden voor meer algemene zaken zoals het bijhouden op welk scherm er gewerkt wordt e.d.

4. Literatuur.

IK hoop dat ik met deze publicatie een aantal mensen heb overgehaald om FORTH te leren en te gaan gebruiken. Om FORTH te leren kan gebruik worden gemaakt van de volgende boeken:

- A. Winfield: Flitsend FORTH, uitg. Academic Service
ISBN 96 6233 1157 (nederlands)
Drs. F.J. Meijer en F. Lemaire: FORTH OK
uitg. Wolfskamp
ISBN 90 70556 0607 (nederlands)
L. Brodie: Starting FORTH, uitg. Prentice-Hall Inc.
ISBN 0 13 842922-7 (engels)

FORTH voor de JUNIOR is te verkrijgen via de redactie-secretaris: W.L. van Pelt, Jacob Jordaanstraat 15
2923 CK Krimpen a.d. IJssel. Tel. 01807 - 19881.

> ASS L

```

0010      LS
0020      : ***** COORDINATEN BIJ USURPATOR II *****
0030      : ***** SCHAAKPROGRAMMA *****
0040
0050      Benodigd : Elektuur JUNIOR-computer met
0060          interfacekaart en monitor.
0070
0080      Auteur   : Gamiel de Lv
0090          E. Blaesstraat 7
0100          B - 9218 GENT. Belgie.
0110
0120      Zie ook  : DE 6502 KENNER 8(1984)1(FEB)8
0130          DE 6502 KENNER 8(1984)4(AUG)22-24, 32
0140
0150          USURPATOR II is een uitoave van
0160          uitgeverij WOLFKAMP te Amsterdam.
0170          Bestelnummer: ISBN 90-70556-06-5
0180
0190      De aanpassing welke Dirk M. Verzienden in DE
0200      6502 KENNER nr. 33 van augustus 1984, oad. 22
0210      en 23 bekendmaakte vond ik fijn. Toch moeten
0220      er ergens een paar fouties inadeslopen zijn.
0230      De onderste stukken komen niet juist op scherm
0240      en de computer geeft geen tekenzettingen. Ik heb
0250      de zaak als volgt oopgelost:
0260      3436 - 4C 3C 22
0270      34B6 - 00
0280      34B7 - 2A 50 4C
0290      34BA - 54 44 4B
0300      34BD - 3A 00 00
0310
0320      De onderstaande aanvulling geeft rond het
0330      schaakkord coordinaten (van 0 to 8 en van
0340      A to H), zoals men in schaakboeken ziet.
0350      Het intoetsen van de eiden zetten is hier-
0360      door eenvoudiger.
0370      Eerst moet het oorspronkelijke programma
0380      waarin de patches uit DE 6502 KENNER nr. 30
0390      en 33 zijn aangebracht, nog worden gewijzigd
0400      als volgt:
0410      3426 - D0 67 was D0 DF
0420      3428 - 4C F0 34 was 20 E8 11
0430      348F - 4C E2 34 was EA EA EA
0440
0450      Monitor routines
0460
0470      CRLF      : carriage return & linefeed
0480      PRCHA    : print character on screen
0490      PRSP     : print a space
0500
0510      BA $34E2
0520
0530      COOR
0540      NUMM
0550
0560
0570
0580      CIJF
0590
0600
0610
0620
0630
0640
0650      LETT
0660
0670
JSR PRSP      : print cijfer
LDA #'8
JSR PRCHA
DEC $34E6      : verminder cijfer
JMP STORE
JSR PRSP
LDA #'1      : print 1
JSR PRCHA
LDA #'8      : zet 38 terug
STA $34E6
JSR CRLF      : 2 x naar rechts
JSR CRLF
LDA #'A      : print letters
JSR PRCHA
JSR PRSP      : 2 x dalen

```

```

350B- 20 F3 11 0680      JSR PRSP
350E- EE 04 35 0690      INC $3504   : nieuwe letter
3511- AD 04 35 0700      LDA $3504   : is H al geprint ?
3514- C9 49 0710      CMP #$49
3516- DO EB 0720      BNE LETT   : neen
3518- A9 41 0730      LDA #'A    : ja, zet 41 terug
351A- BD 04 35 0740      STA $3504
351D- 20 E8 11 0750      JSR CRLF
3520- 4C 2B 34 0760      JMP WIT/ZWART?
                                :
                                0770
                                0780 WIT/ZWART? . DE $342B
                                0790 STORE  . DE $3407
                                0800
                                0810      : W.L. van Pelt: Do eenvoudige wijze kan
                                0820      : het schaakprogramma nog zodanig gewij-
                                0830      : ziod worden dat we het omhoog scrollen
                                0840      : van het schaakk bord tegemoetgaan, door het
                                0850      : inbouwen van een CLSR met delay van 132
                                0860      : mSec i.v.m. de tijd die een monitor kan
                                0870      : vragen om het scherm schoon te maken.
                                0880      : Er wordt gebruik gemaakt van de timer.
                                0890
                                0900 RDFLAG  . DE $1AD5   : flag register
                                0910 CNTD     . DE $1AF7   : count 1024 T
                                0920
                                0930      : BA $22FA
                                0940
                                0950      JMP CLSR
                                0960
                                0970      : BA $3570
                                0980
                                0990 CLSR   PHA :      : bewaar inhoud akku
                                1000      LDA #$0C   : zend form feed naar monitor
                                1010      JSR PRCHA
                                1020      LDA #$84   : wachtloop 132 mSec
                                1030      STA CNTD
                                1040 CA     BIT RDFLAG
                                1050      BPL CA
                                1060      PLA
                                1070      JMP $3400   : en terug
                                1080 ;


//0000, 3584, 3584
) RUN $16D5

```

TE KOOP:

JUNIOR Computer + interface + 24 K Ram + Elekterminal
+ keyboard + boeken 1, 2, 3 en 4, en de Intelekt schaak-computer, alles goed werkend in een kooi: fl. 1000,-.
M. van Nunspeet. Tel. Werk: 043 - 478208
Tel. thuis: 043 - 473561

OHIO DOS V3.3

Als je bestaande routines in machinetaal aan wilt passen van de standaard JUNIOR (of andere) naar de DOS-versie dan zijn de volgende adressen/routines van belang:

Standaard	DOS	Routine	
\$12AE	\$2340	RECCHA met echo	Deze routines kijken welk in/out device ingesteld is, waarna de ingedrukte waarde in \$2363 komt te staan.
	\$2336	RECCHA zonder echo	
\$1334	\$2343	PRCHA	Deze routine zet zelf de waarde van accu naar \$2363, waarna hij uitgeprint wordt volgens het ingestelde output device (beeldscherm en/of printer).

\$128F \$2D92 PRBYTE
Wout van Dinther, Molenrijnselaan 27, 5262 TN VUGHT.

HOW TO CHANGE THE MEMORY MAP OF YOUR FORTH SYSTEM ON JUNIOR

By: Gert van Dobroek, Hooplanden 20, 9801 LB Zuidhorn (NL)

The FORTH from the cassette library for the JUNIOR-computer uses only RAM-addresses less than \$A000. When your system has more RAM you can adapt FORTH by the following changes:

Calculate the new values of:

ERAM	: End-of-RAM+1
MEM	= ERAM - \$4000 : Begin of tape buffer (RAM-disk)
UAREA	= MEM - \$ 80 : Begin of user area
DAREA	= UAREA - \$ 420 : Begin of disk buffers

Put ERAM in location : \$34B8
Put MEM in locations: \$34B0, \$3899
Put UAREA in locations: \$2010, \$26A2
PUT DAREA in locations: \$2696, \$31C4, \$31CF

Change the call of U((old) in CREATE in the call of U((new) by the batch: \$2E41 was \$2813 becomes \$3F0D.

CAUTION: After these changes, old words written to tape by the word TO-TAPE can no longer be loaded.

On page 21 of DE 6502 KENNER nr. 34 my solution on recursive was not published well. It ought to be as follows:

: MYSELF LATEST PFA CFA, : IMMEDIATE
: RECURSIE ." 6502 " R) DRDP (don't let the return-stack)
(flow over)

MYSELF :

Another example:

: RECFAC (compute n! recursive)
DUP 1) 1F DUP 1- MYSELF * ENDIF :

Your call: n RECFAC

NOTULEN VAN DE ALGEMENE JAARVERGADERING DD. 24-11-1984

Uitreiking publikatie-aanmoedigingspremie(s)

Namens de jury maakt Anton Mueller bekend dat de tweede prijs is toegekend aan Piet Fransen voor zijn publicatie over de koppling van de JUNIOR aan het Proton DOS. De eerste prijs wordt toegekend aan Koen van Nieuwenhove in verband met zijn publicatie over de omzetting van een 8080 DOS naar 6502 assembler en de implementatie daarvan op de JUNIOR.

Opening van de vergadering

De voorzitter heet de aanwezigen welkom, maar betreurt het dat de opkomst laag is. Het ligt in de bedoeling in de toekomst de jaarvergadering in het begin van het jaar te gaan houden. Dit is praktischer in verband met de bespreking van het financieel jaarverslag en de begroting van het volgend jaar.

Notulen jaarvergadering dd. 19-11-1983

Er zijn geen opmerkingen, het verslag wordt door de vergadering aangenomen.

Financieel jaarverslag 1983

De voorzitter memoreert dat Riet Uphoff, ondanks het feit dat zij haar functie als penningmeester per 01-01-1984 had neergelegd, toch bereid is geweest dit jaar de financien te beheren. Het boekjaar 1983 blijkt met een groot positief saldo te zijn afgesloten.

De vergadering vraagt zich af hoe het grote verschil tussen de portikosten voor de verzending van de 6502 KENNER en overige portikosten te verklaaren is. Het bestuur verklaart dat met name de toegenomen aktiviteiten van de redactie (paperwareservice, cassettes, propaganda) tot een stijging van de kosten hebben geleid. Verder zijn in deze post ook de portikosten van het secretariaat en de overige bestuursleden inbegrepen. Men dient overigens wel te bedenken dat tegenover de uitgaven ook inkomsten bestaan, met name van de redactie. Daarnaast heeft de club zich vrij intensief beziggehouden met propaganda om het ledental op peil te houden.

In het verslag blijkt een foutje geslopen te zijn in de boeking van de rente op de girorekening van de club. De rente blijkt op de post reservefonds te zijn geboekt.

Van de kascontrolecommissie heeft het bestuur geen verslag mogen ontvangen. Wel heeft de commissie het kasboek voor accoord geparafeerd. De commissie wordt voor zijn diensten bedankt.

Het bestuur deelt mede dat naar haar mening de boeking van de contributies niet juist is gebeurd. Het is niet meer na te gaan wanneer de contributies voor welk jaar zijn voldaan. Als gevolg hiervan doet het bestuur het voorstel om de boekhouding door een accountant te laten controleren. Dat geeft de nieuwe penningmeester de mogelijkheid om met een schone lei te beginnen. Dit voorstel is er ook de oorzaak van dat het bestuur geen begroting voor 1984 en 1985 heeft gemaakt. Na afkomst van het accountantsrapport zal dit alsnog voor 1985 gebeuren. Begin 1985 zal deze begroting aan de ledenvergadering worden voorgelegd. De heer Kan, lid van onze vereniging en accountant van beroep, zal gevraagd worden dit rapport samen te stellen. De vergadering

DE 6502 KENNER

gaat akkoord met dit voorstel.

De vergadering doet het voorstel de club KIM (en eventueel ook de club JUNIOR) tegen restwaarde van de hand te doen. Het bestuur zal dit in overweging nemen.

De heren W. Cuypers en H.C. van Tiel zullen plaats nemen in de nieuwe kascontrolecommissie.

Verkiezing nieuwe bestuursleden

Ruud Uphoff heeft te kennen gegeven dat hij in verband met drukke werkzaamheden het secretariaat zal neerleggen. De vergadering bedankt hem voor bewezen diensten. Op voordracht van het bestuur wordt Gert Klein gekozen tot nieuwe secretaris. Eveneens op voordracht van het bestuur wordt John van Sprang, nu reeds gewoon bestuurslid, tot penningmeester gekozen. Het bestuur verzoekt om een kandidaat voor de vacante zetel van John van Sprang. Nico de Vries meldt zich aan en wordt gekozen.

Rondvraag

De vergadering geeft de redactie een compliment voor de uitstekende wijze waarop de 6502 KENNER wordt verzorgd.

De vraag wordt gesteld of er binnen de vereniging wel voldoende aandacht wordt besteed aan de Commodore 64 bezitters die lid willen worden van de club. De voorzitter antwoordt dat het bestuur er zich van bewust is dat we tot nu toe te weinig aandacht hebben besteed aan de opvang van nieuwe leden. Anton Mueller heeft zich inmiddels bereid verklaart in de 6502 KENNER een artikelenserie te gaan plaatsten die met name op de beginnende leden binnen de club gericht zal zijn. Daarnaast zullen een aantal ervaren leden zich intensief met het C64 gebeuren gaan bezighouden. Uitgangspunt daarbij blijft wel dat kennisuitwisseling op hoog niveau de doelstelling van de club is.

De vergadering vraagt of de verlaging van de toegangsprijzen voor de bijeenkomsten tot een verhoogd aantal bezoekers heeft geleid. De voorzitter verklaart dat dit niet merkbaar is. Er blijkt op de bijeenkomsten meestal een vaste kern van leden te komen. Het is niet te zeggen of de club op dit moment geld moet toeleggen op de bijeenkomsten. Het bestuur houdt zich van harte aanbevolen voor gratis locaties.

Op verzoek van de vergadering zegt de redactie toe bij de uitnodigingen voor de bijeenkomsten een situatiekaartje te publiceren.

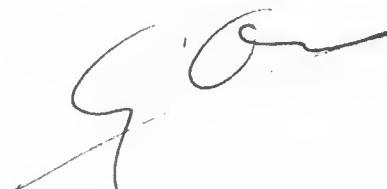
Op het verzoek om bij listings in de 6502 KENNER beter aan te geven welke delen gewijzigd moeten worden, om een programma op een ander systeem te laten draaien, antwoordt de redactie dat hieraan wel meer aandacht besteed zal worden, maar dat de redactie hiervoor ook afhankelijk is van de auteurs van deze programma's. Het probleem heeft echter de aandacht.

Sluiting

De voorzitter sluit de vergadering.

Wageningen, 25-11-1984

Gert Klein



M2

BALANS PER 31 DECEMBER 1984

AKTIVA			PASSIVA			
	1984	1983			1984	1983
Inventaris	3950	6770	Vrije reserve		13432	11710
Voorraden	PM	PM	Reservering zesde			
Te ontvangen			uits 6502 kenner			
Posten	519	505	Vooruitontvangen		2400	2400
Geldmiddelen	24007	20056	contributie van			
			272 cq 282 leden			
			Te betalen posten		12240	12690
			404		531	
	28476	27331				
					28476	27331

STAND VAN BATEN EN LASTEN OVER 1984.

LASTEN.			BATEN			
	1984	1983			1984	1983
Drukkosten 6502 kenner	13027	11124	Contributie van			
Verzend kosten	1093	950	438 cq 409 leden		19710	18405
TOTAAL	14120	12074	Reclame		920	300
Af reservering zesde uitslagen	2400	-	Verkoop van 6502			
TOTAAL	11720	12074	kenner		348	272
Nagekomen verzend kosten uit 1982	-	201	Redactie inkomsten		3351	3208
Redactie kosten	2611	2327	Bijeenkomsten		888	1316
Bestuurs Kosten	2203	2414	Interest bank, giro		502	529
HCC dag(en)	2157	763				
Afschrijving Inventaris	2820	2460				
TOTAAL UITGAVEN	21597	20239				
Te besteden batingsaldo						
Reservering zesde						
uits 6502 kenner	2400	2400				
Toevoeging aan vrije						
reserve	1722	1891				
TOTAAL BATINGSALDO	4122	4291				
TOTAAL	25719	24530				
					25719	24530

INVENTARIS

		aanschaffings waarde	afschrijvingen t/m 1983	1984	boekwaarde 31-12-84
schrijfmachine	1981	1224	604	310	310
Printer oki-80	1981	590	300	150	140
tractorfeed	1981	175	45	40	90
apple II	1982	4000	1800	1000	1200
kim computer	1980	1294	914	320	60
junior computer	1981	399	209	100	90
flipover	1980	310	200	80	30
twee recorders	1982	394	94	100	200
Printer oki-84	1983	2613	323	650	1640
monitor	1983	299	39	70	190
		11298	4528	2820	3950

Het afschrijvingspercentage bedraagt 25% van de aanschaffingswaarde.

BEGROTING 1985

LASTEN

Druk kosten 6502 kenner	
en verzend kosten 6x	
Bestuurs Kosten	14400
Redactie Kosten	2500
HCC Dagen	2500
Inventaris	1500
	2500
	23400

BATEN

Contributie	
a f45 400 leden	18000
Opbrengst 6502	
kenner	200
Bijeenkomsten	
4x inkomsten	500
Reclame	1200
Redactie	3000
Interest bank, giro	500
	23400

J. F. v. Sprang
Penningmeester.

PMDL2 MICRO-WARE ASSEMBLER 65XX-1.1

PAGE 01

```

0010:          GREEDY DEEL 2
0020:
0030:          PHONS BLOEMEN
0040:          VINKELAAN 200
0050:          5702 LX HELMOND
0060:
0070: 8C00      PMDL2 ORG $8C00
0080:
0090:          *** KLABBLOK-RAM ***
0100:
0110: 02 00 VELDPL * $0002 VECTOR IN VELDGEGEVENS RAM
0120: 04 00 PRPNTL * $0004 PRINT-VECTOR
0130: 08 00 PMAANT * $0008 AANTAL GREEDIES
0140: 09 00 EATFLG * $0009 ETEN / GEGETEN WORDEN
0150: 0A 00 TIMLFT * $000A AANTAL VAKJES OVER
0160: 0B 00 JMIL * $000B GEBRUIKT VOOR JMI
0170: 0D 00 DIREC * $000D SPEELRICHTING
0180: 0E 00 DIRPM * $000E LOOPRICHTING GREEDY
0190: 10 00 PMFLAG * $0010 VLAGGEN DIE AANGEVEN OF
0200: 18 00 PMTIM * $0018 VERTRAGINGSTIJDBUFFERS
0210: 20 00 HCOORL * $0020 HULPCOORDINATEN I
0220: 22 00 MCOORL * $0022 HULPCOORDINATEN II
0230: 24 00 PMPOS1 * $0024 POSITIE VAN GREEDY
0240: 26 00 MAPOS1 * $0026 POSITIE MONSTER A
0250: 30 00 PLUSB * $0030 OPTELBUFFER (4 BYTES)
0260: 34 00 STARPT * $0034 AANTAL PUNTEN PER STERRETJE
0270: 50 00 EXTRA * $0050 AANTAL EXTRA GEKREGEN GREEDIES
0280: 52 00 DIRECA * $0052 RICHTINGSBYTE VOOR DIREC.
0290: 53 00 CTRLA * $0053 CONTROLBYTES
0300: 58 00 PMSPED * $0058 SNELHEID GREEDY
0310: 59 00 MONNUM * $0059 AANTAL MONSTERS IN SPEL
0320: 60 00 MASTAT * $0060 STATUS MONSTER A
0330: 6A 00 MADIR * $006A LOOPRICHTING MONSTER A
0340:
0350:          *** I/O, TIMER, IRQ, VIDEO ***
0360:
0370: FD 1A CLKKTI * $1AFD 8 uS TIMER, IRQ
0380: 80 1A PAD * $1A80 POORT A
0390: 81 1A PADD * $1A81 RICHTING POORT A
0400: 82 1A PBD * $1A82 POORT B
0410: 83 1A PBDD * $1A83 RICHTING POORT B
0420:
0430:          *** PLAATSEN IN VIDEO RAM ***
0440:
0450: 96 D2 PMRES * $D296 RESERVE GREEDIES
0460: F8 D2 SCORD * $D2F8 SCORE
0470: F8 D3 TEXTD * $D3F8 BEGIN VAN TEKSTKADER
0480: 7B D5 PUNTD * $D57B AANTAL PUNTJES NOG OVER
0490: FA D6 BONUSD * $D6FA BONUS OP EINDE VAN STAGE
0500: 00 D7 TIMBAL * $D700 TIJDHALF
0510:
0520: 70 88 NXTSTA * $8870
0530: 3D 89 PCRT * $893D
0540: 65 89 PRINT * $8965
0550: BA 89 CLSCR * $89BA
0560: D5 89 PMMORS * $89D5
0570: 35 8A PMFIN * $8A35
0580: B1 8A MONRES * $8AB1
0590: C2 8A RESMON * $8AC2
0600: F9 8A PMATTK * $8AF9
0610: 30 8B TXTSET * $8B30
0620: 39 8B TXTSHW * $8B39
0630: 58 8B DELAYA * $8B58
0640: 63 8B TXTSOW * $8B63

```

PMDL2 MICRO-WARE ASSEMBLER 65XX-1.1

PAGE 02

```

0650: 6E 8B PMDIS * $8B6E
0660: 91 8B DECBAL * $8B91
0670: B1 8B DISVEC * $8BB1
0680: BA 8B VELVEC * $8BBA
0690:
0700: DEEL 2
0710:
0720: *** PLAYER ROUTINE ***
0730:
0740: 8C00 20 38 8E PMROUT JSR PLAYER HAAL ZET SPELER OP
0750: 8C03 A5 24 LDA PMPOS1
0760: 8C05 85 20 STA HCOORL
0770: 8C07 A5 25 LDA PMPOSL +01 KIJK OF PM
0780: 8C09 85 21 STA HCOORL +01 WEL DE GEVRAAGDE
0790: 8C0B 20 BA 8B JSR VELVEC KANT UIT KAN
0800: 8C0E A5 0D LDA DIREC (- GEVRAAGDE RICHTING
0810: 8C10 A0 00 LDYIM $00 KIJK OF HET KAN
0820: 8C12 31 20 ANDIY HCOORL VERGELIJK
0830: 8C14 D0 06 BNE POSSIB 0 = KAN NIET
0840: 8C16 A5 52 LDA DIRECA PAK OUDE RICHTING
0850: 8C18 31 20 ANDIY HCOORL VERGELIJK
0860: 8C1A F0 23 BEQ PMEND KAN HET NIET ?
0870: 8C1C 85 52 POSSIB STA DIRECA DIT WORDT DE NIEUWE RICHTING
0880: 8C1E 20 B1 8B JSR DISVEC VOER DEZE ZET UIT
0890: 8C21 20 83 8E JSR MOVE
0900: 8C24 20 BA 8E JSR TUNCHK LET OP TUNNELS
0910: 8C27 20 EB 8E JSR RESPLU SCORE OP 0
0920: 8C2A B1 20 LDAIY HCOORL KIJK OF ER IS GESCOORD
0930: 8C2C C9 21 CMPIM $21 LEEG VELD ?
0940: 8C2E 90 0C BCC SCREEN
0950: 8C30 A2 00 LDIXIM $00 GA SCORETABEL AF
0960: 8C32 DD 4A 8C CHECK CMPAX PUNTAB
0970: 8C35 F0 3D BEQ YES SPRING ALS ER GESCOORD IS
0980: 8C37 E8 INX ZO NIET. PAK VOLGENDE
0990: 8C38 E0 0E CPXIM $0E ALLES GEHAD ?
1000: 8C3A D0 F6 BNE CHECK
1010: 8C3C 20 6E 8B SCREEN JSR PMDIS ZET ZET OP SCHERM
1020: 8C3F A9 00 PMEND LDAIM $00 RESET DE VLAG
1030: 8C41 85 10 STA PMFLAG
1040: 8C43 A5 58 LDA PMSPED ZET TIJDBUFFER
1050: 8C45 85 18 STA PMTIM
1060: 8C47 4C 35 8A JMP PMFIN GA TERUG NAAR HOOFDLUS
1070:
1080: 8C4A AB PUNTAB = $AB PUNT : 3
1090: 8C4B 2E = $2E DUNNE PUNT : 7
1100: 8C4C 2A = $2A STERRETJE : 10.12.15.18. ETC
1110: 8C4D EA = $EA DRIEHOEK : 12
1120: 8C4E EB = $EB DRIEHOEK : 12
1130: 8C4F E2 = $E2 POWER : 15
1140: 8C50 B7 = $B7 BARRIERE. DOE NIETS
1150: 8C51 B8 = $B8 BARRIERE
1160: 8C52 E5 = $E5 MONSTER : PREMIE.
1170: 8C53 E6 = $E6 OF OPGEVREten
1180: 8C54 E7 = $E7
1190: 8C55 E8 = $E8
1200: 8C56 E9 = $E9
1210: 8C57 BB = $BB KWETSBAAR MONSTER
1220: 8C58 81 PUNTAC = POINT
1230: 8C59 D6 = DUNNIE
1240: 8C5A 35 = STER
1250: 8C5B DD = DRIEK
1260: 8C5C DD = DRIEK
1270: 8C5D 43 = POWER
1280: 8C5E 3F = PMEND

```

PMDL2 MICRO-WARE ASSEMBLER 65XX-1.1

PAGE 03

```

1290: 8C5F 3F      =    PMEND
1300: 8C60 F2      =    DEAD
1310: 8C61 F2      =    DEAD
1320: 8C62 F2      =    DEAD
1330: 8C63 F2      =    DEAD
1340: 8C64 F2      =    DEAD
1350: 8C65 7D      =    EATEN
1360: 8C66 8C      PUNTAD =    POINT   /
1370: 8C67 8C      =    DUNNIE  /
1380: 8C68 BD      =    STER    /
1390: 8C69 8C      =    DRIEK   /
1400: 8C6A 8C      =    DRIEK   /
1410: 8C6B BD      =    POWER   /
1420: 8C6C 8C      =    PMEND   /
1430: 8C6D 8C      =    PMEND   /
1440: 8C6E BD      =    DEAD    /
1450: 8C6F BD      =    DEAD    /
1460: 8C70 BD      =    DEAD    /
1470: 8C71 BD      =    DEAD    /
1480: 8C72 BD      =    DEAD    /
1490: 8C73 BD      =    EATEN   /
1500:
1510: 8C74 BD 58 8C YES    LDAAX PUNTAC HAAL JMI ADRES OP
1520: 8C77 85 0B      STA     JAMIL
1530: 8C79 BD 66 8C      LDAAX PUNTAD
1540: 8C7C 85 0C      STA     JAMIL +01
1550: 8C7E 6C 0B 00      JMI     JAMIL EN SPRING
1560:
1570: *** PUNT ROUTINE ***
1580:
1590: 8C81 20 2E 8F      POINT   JSR     DEPOIN 1 PUNT MINDER
1600: 8C84 B0 0A      BCS     LEEG    ALLE PUNTJES WEG ?
1610: 8C86 A9 02      LDAIM   $02    2 PUNTEN
1620: 8C88 85 33      STA     PLUSB  +03
1630: 8C8A 20 F7 8E      OPTEL   JSR     ADD
1640: 8C8D 4C 3C 8C      LEEG    JMP     SCREEN
1650: 8C90 A9 02      LDAIM   $02    2 PUNTEN BIJTELLEN
1660: 8C92 85 33      STA     PLUSB  +03
1670: 8C94 20 F7 8E      JSR     ADD
1680: 8C97 20 6E 88      JSR     PMDIS  LAAT ZIEN
1690: 8C9A A9 01      LDAIM   $01    ZELF LAATSTE PUNTJES OPGEGETEN
1700: 8C9C 85 53      EMPTY   STA     CTRLA
1710: 8C9E 78      SEI     PRINT  GEEN INTERRUPTS
1720: 8C9F 20 30 8B      JSR     TXTSET PRINT TEKST
1730: 8CA2 20 65 89      JSR     PRINT
1740: 8CA5 53      =
1750: 8CA6 54      =
1760: 8CA7 41      =
1770: 8CA8 47      =
1780: 8CA9 45      =
1790: 8CAA 09      =
1800: 8CAB 3C      =
1810: 8CAC 43      =
1820: 8CAD 4F      =
1830: 8CAE 4D      =
1840: 8CAF 09      =
1850: 8CB0 3C      =
1860: 8CB1 50      =
1870: 8CB2 4C      =
1880: 8CB3 45      =
1890: 8CB4 54      =
1900: 8CB5 45      =
1910: 8CB6 44      =
1920: 8CB7 03      =

```



CURSOR

```

1930: 8C88 20 39 8B      JSR    TXTSHW LAAT EVEN ZIEN
1940: 8CBB A5 53      LDA    CTRLA
1950: 8CBD F0 05      BEQ    MONET ZELF LAATSTE PUNNTJES OPGEGETEN ?
1960: 8CBF A9 25      LDAIM $25   25 X DE BONUS
1970: 8CC1 4C C6 8C      JMP    BONPLU
1980: 8CC4 A9 07      MONET LDAIM $07   7 X DE BONUS
1990: 8CC6 20 71 8F      BONPLU JSR    BONBER
2000: 8CC9 A2 0A      LDXIM $0A   ZET CURSOR UIT
2010: 8CCB A9 20      LDAIM $20
2020: 8CCD 20 3D 89      JSR    PCRT
2030: 8CD0 20 B1 8A      JSR    MONRES ZET MONSTERS TERUG
2040: 8CD3 4C 70 88      JMP    NXTSTA VOLGEND STAGE
2050:
2060: 8CD6 A9 03      DUNNIE LDAIM $03   3 PUNTEN
2070: 8CD8 85 33      STA    PLUSB +03
2080: 8CDA 4C 8A 8C      JMP    OPTEL
2090:
2100: *** SLEUTEL ROUTINE ***
2110:
2120: 8CDD A5 20      DRIEK  LDA    HCOORL HAAL COORDINATEN OP
2130: 8CDF 4A      LSRA   DIT ZIJN D052..D062 <1>, <2>
2140: 8CE0 4A      LSRA   D282..D2B2 <0>, <3>
2150: 8CE1 4A      LSRA   D402..D432 <4>, <7>
2160: 8CE2 4A      LSRA   D652..D662 <5>, <6>
2170: 8CE3 29 03      ANDIM $03   BEWERK ZE EEN BEETJE
2180: 8CE5 AA      TAX    TOT ZE EEN ACCEPTABEL
2190: 8CE6 A5 21      LDA    HCOORL +01 NUMMER HEBBEN
2200: 8CE8 29 04      ANDIM $04
2210: 8CEA D0 04      BNE    BREAK
2220: 8CEC 8A      TXA    ONERSTE HELFT: <4. 5. 6, 7>
2230: 8CED 09 04      ORAIM $04
2240: 8CEF AA      GETCOR TAX
2250: 8CF0 BD 15 8D      BREAK LDAAX BLOKH HAAL COORDINATEN OP
2260: 8CF3 85 23      STA    MCOORL +01 VAN BLOKKADE
2270: 8CF5 BD 25 8D      LDAAX BLOKL
2280: 8CF8 85 22      STA    MCOORL
2290: 8CFA A0 00      LDYIM $00
2300: 8CFC A9 20      LDAIM $20
2310: 8CFE 91 22      STAIY MCOORL EN RUIM HEM OP
2320: 8D00 8A      TXA    2 BLOKKADES PER DRIEHOEK
2330: 8D01 C9 08      CMPIM $08   ALLEBEI AL GEHAD ?
2340: 8D03 10 05      BPL    BRUP
2350: 8D05 09 08      ORAIM $08
2360: 8D07 4C EF 8C      JMP    GETCOR
2370: 8D0A A9 02 8C      BRUP  LDAIM $02   12 PUNTEN
2380: 8D0C 85 33      STA    PLUSB +03
2390: 8D0E A9 01      LDAIM $01
2400: 8D10 85 32      STA    PLUSB +02
2410: 8D12 4C 8A 8C      JMP    OPTEL
2420:
2430: 8D15 D0      BLOKH = $D0
2440: 8D16 D1      = $D1
2450: 8D17 D1      = $D1
2460: 8D18 D0      = $D0
2470: 8D19 D5      = $D5
2480: 8D1A D5      = $D5
2490: 8D1B D5      = $D5
2500: 8D1C D5      = $D5
2510: 8D1D D2      = $D2
2520: 8D1E D3      = $D3
2530: 8D1F D3      = $D3
2540: 8D20 D2      = $D2
2550: 8D21 D4      = $D4
2560: 8D22 D3      = $D3

```

PMDL2 MICRO-WARE ASSEMBLER 65XX-1.1

PAGE 05

2570: 8D23 D3	=	\$D3	
2580: 8D24 D4	=	\$D4	
2590: 8D25 CA	BLOKL	=	\$CA
2600: 8D26 82		=	\$82
2610: 8D27 B2		=	\$B2
2620: 8D28 EA		=	\$EA
2630: 8D29 CA		=	\$CA
2640: 8D2A 02		=	\$02
2650: 8D2B 32		=	\$32
2660: 8D2C EA		=	\$EA
2670: 8D2D 12		=	\$12
2680: 8D2E 0E		=	\$0E
2690: 8D2F 26		=	\$26
2700: 8D30 22		=	\$22
2710: 8D31 92		=	\$92
2720: 8D32 8E		=	\$8E
2730: 8D33 A6		=	\$A6
2740: 8D34 A2		=	\$A2
2750:			
2760:	*** BONUS EN POWER ***		
2770:			
2780: 8D35 A2 00	STER	LDXIM \$00	HAAL PUNTENAANTAL OP
2790: 8D37 B5 34	STERA	LDAAX STARPT	
2800: 8D39 95 30		STAAX PLUSB	EN ZET IN OPTELBUFFER
2810: 8D3B E8		INX	
2820: 8D3C E0 04		CPXIM \$04	
2830: 8D3E D0 F7		BNE STERA	
2840: 8D40 4C 8A BC		JMP OPTEL	
2850:			
2860: 8D43 A9 35	POWER	LDAIM \$35	ZET BLOKTELLER OP 53
2870: 8D45 85 0A		STA TIMLFT	
2880: 8D47 AA		TAX	
2890: 8D48 CA	WIT	DEX	
2900: 8D49 30 08		BMI WHITE	
2910: 8D4B A9 A1		LDAIM \$A1	MAAK TIJDBALK WIT
2920: 8D4D 9D 00 D7		STAAX TIMBAL	
2930: 8D50 4C 48 BD		JMP WIT	
2940: 8D53 A9 01	WHITE	LDAIM \$01	ZET EETVLAG
2950: 8D55 85 09		STA EATFLG	
2960: 8D57 A2 00		LDXIM \$00	
2970: 8D59 B5 60	KWETS	LDAAX MASTAT	MAAK ALLE MONSTERS
2980: 8D5B 09 80		ORAIM \$80	KWETSBAAR
2990: 8D5D 95 60		STAAX MASTAT	
3000: 8D5F B5 6A		LDAAX MADIR	EN KEER HUN RICHTING OM
3010: 8D61 20 D8 8F		JSR TURN	
3020: 8D64 95 6A		STAAX MADIR	
3030: 8D66 E8		INX	
3040: 8D67 E0 05		CPXIM \$05	
3050: 8D69 D0 EE		BNE KWETS	
3060: 8D6B A2 0A		LDXIM \$0A	ZET CURSOR OP KNIPPEREN
3070: 8D6D A9 40		LDAIM \$40	
3080: 8D6F 20 3D 89		JSR PCRT	
3090: 8D72 A9 05		LDAIM \$05	15 PUNTEN
3100: 8D74 85 33		STA PLUSB	+03
3110: 8D76 A9 01		LDAIM \$01	
3120: 8D78 85 32		STA PLUSB	+02
3130: 8D7A 4C 8A BC		JMP OPTEL	
3140:			
3150:	*** MONSTER OPGEGETEN ***		
3160:			
3170: 8D7D A2 00	EATEN	LDXIM \$00	MONSTER DOOR GREEDY OPGEGETEN
3180: 8D7F A5 20	EATENA	LDA HCOORL	BEPAAI WELK
3190: 8D81 D5 26		CMPAX MAPOS	GREEDY STAAT IN HCOORL
3200: 8D83 D0 06		BNE EATENB	

DE 6502 KENNER

PMDL2 MICRO-WARE ASSEMBLER 65XX-1.1

PAGE 06

```

3210: 8D85 A5 21      LDA    HCOORL +01
3220: 8D87 D5 27      CMPAX  MAPOS1 +01
3230: 8D89 F0 09      BEQ    EATENC MONSTER BEPAALD ?
3240: 8D8B E8          EATENB INX    2 KEER !
3250: 8D8C E8          INX
3260: 8D8D E0 0A      CPXIM $0A    ALLE MONSTERS GEHAD ?
3270: 8D8F D0 EE      BNE    EATENA
3280: 8D91 4C 3C 8C      JMP    SCREEN HET WAS EEN VLEKJE
3290: 8D94 20 9A 8D      EATENC JSR    MDOWN VOER OPEETROUTINE UIT
3300: 8D97 4C 3F 8C      JMP    PMEND EN GA DOOR
3310:
3320: *** MONSTER OPEET SUBROUTINE ***
3330:
3340: 8D9A 8A          MDOWN TXA    BETREFFENDE MONSTER
3350: 8D9B 4A          LSRA
3360: 8D9C AA          TAX
3370: 8D9D 78          SEI    GEEN INTERRUPTS
3380: 8D9E B5 60      LDAAX MASTAT MAAK HET WEER ONKWETSBAAR
3390: 8DAO 29 0F      ANDIM $0F
3400: 8DA2 95 60      STAAX MASTAT BIT 7 =0, BIT 6 =1
3410: 8DA4 20 C2 8A      JSR    RESMON STUUR HET NAAR START
3420: 8DA7 20 6E 8B      JSR    PMDIS EN ZET DE GREEDY EROP
3430: 8DAA 20 30 8B      JSR    TXTSET
3440: 8DAD 20 65 89      JSR    PRINT
3450: 8DB0 09          =
3460: 8DB1 40          =    $40    SLA REGEL OVER
3470: 8DB2 53          =
3480: 8DB3 50          =
3490: 8DB4 4F          =
3500: 8DB5 4F          =
3510: 8DB6 4B          =
3520: 8DB7 03          =
3530: 8DB8 A9 04      LDAIM $04
3540: 8DBA A6 09      LDX    EATFLG BEPAAL HOEVEELSTE MONSTER
3550: 8DBC F0 1C      BEQ    OPTIM (ERROR)
3560: 8DBE A9 04      LDAIM $04
3570: 8DC0 CA          DEX
3580: 8DC1 F0 0C      BEQ    MONBON WORDT OPGEGETEN
3590: 8DC3 A9 08      LDAIM $08    1E : 4 X BONUS
3600: 8DC5 CA          DEX    2E : 8 X BONUS
3610: 8DC6 F0 07      BEQ    MONBON 3E : 16 X BONUS
3620: 8DC8 A9 16      LDAIM $16    4E. 5E 32 X BONUS
3630: 8DCA CA          DEX
3640: 8DCB F0 02      BEQ    MONBON
3650: 8DCD A9 32      LDAIM $32
3660: 8DCF 20 71      8F    MONBON JSR    BONBER
3670: 8DD2 A5 09      LDA    EATFLG KIJK OF ALLE MONSTERS OP ZIJN
3680: 8DD4 C5 59      CMP    MONNUM
3690: 8DD6 F0 04      BEQ    ALLUP
3700: 8DD8 E6 09      INC    EATFLG BONUS VOOR VOLGEND MONSTER
3710: 8DDA 58          OPTIM CLI    INTERRUPTS MOGELIJK
3720: 8DDB 60          RTS
3730: 8DDC A6 0A      ALLUP LDX    TIMLFT HAAL SEGMENTENTELLER
3740: 8DDE E0 01      CPXIM $01 VOOR GOEDE AFWIKKELING
3750: 8DE0 F0 F8      BEQ    OPTIM ALLES AL WEG ?
3760: 8DE2 20 EB 8E      JSR    RESPLU OPTELBUFFER 0000
3770: 8DE5 A9 04      LDAIM $04 4 PUNTEN VOOR ELK SEGMENT
3780: 8DE7 85 33      STA    PLUSB +03 DAT NOG OVER IS
3790: 8DE9 20 F7 8E      JSR    ADD TEL OP
3800: 8DEC 20 91 8B      JSR    DECBAL EEN SEGMENT MINDER
3810: 8DEF 4C DC 8D      JMP    ALLUP
3820:
3830: *** GREEDY OPGEGETEN ***
3840:

```

PMDL2 MICRO-WARE ASSEMBLER 65XX-1.1

PAGE 07

3850: 8DF2 78	DEAD	SEI	GEEN INTERRUPTS MEER
3860: 8DF3 20 52 8F		JSR CURSET	MONSTER OP GREEDY
3870: 8DF6 20 30 8B		JSR TXTSET	
3880: 8DF9 20 65 89		JSR PRINT	
3890: 8DFA 45		= 'E	
3900: 8DFD 41		= 'A	
3910: 8DFF 54		= 'T	
3920: 8DFF 45		= 'E	
3930: 8E00 4E		= 'N	
3940: 8E01 09		= \$09	
3950: 8E02 3D		= \$3D	
3960: 8E03 42		= 'B	
3970: 8E04 59		= 'Y	
3980: 8E05 09		= \$09	
3990: 8E06 3C		= \$3C	
4000: 8E07 47		= 'G	
4010: 8E08 48		= 'H	
4020: 8E09 4F		= 'O	
4030: 8E0A 53		= 'S	
4040: 8E0B 54		= 'T	
4050: 8E0C 03		= \$03	
4060: 8E0D 20 39 8B		JSR TXTSHW	LAAT EVEN ZIEN
4070: 8E10 C6 08		DEC PMAANT	1 GREEDY MINDER
4080: 8E12 30 03		BMI GAMOV	ALLES OP?
4090: 8E14 4C D5 89	GAMOV	JMP PMMORS	START OPNIEUW
4100: 8E17 20 30 8B		JSR TXTSET	
4110: 8E1A 20 65 89		JSR PRINT	
4120: 8E1D 20		= ,	
4130: 8E1E 47		= 'G	
4140: 8E1F 41		= 'A	
4150: 8E20 4D		= 'M	
4160: 8E21 45		= 'E	
4170: 8E22 09		= \$09	
4180: 8E23 7C		= \$7C	VOLGENDE REGEL
4190: 8E24 4F		= 'O	
4200: 8E25 56		= 'V	
4210: 8E26 45		= 'E	
4220: 8E27 52		= 'R	
4230: 8E28 03		= \$03	
4240: 8E29 A9 40		LDAIM \$40	WACHT EXTRA LANG
4250: 8E2B 20 3B 8B		JSR TXTSHW	+02 LABEL DELAYA
4260: 8E2E A2 0A		LDXIM \$0A	CURSOR UIT
4270: 8E30 A9 00		LDAIM \$00	
4280: 8E32 20 3D 89		JSR PCRT	
4290: 8E35 4C 00 90		JMP NMWR	GA NAAR HISCORE ROUTINE
4300:		*** SUBROUTINES ***	
4310:		PLAYER : VOERT ZET SPELER IN	
4320:			
4330:			
4340:			
4350: 8E38 A9 1E	PLAYER	LDAIM \$1E	ZET I/O OP
4360: 8E3A 8D 83 1A		STA PBDD	= 00011110
4370: 8E3D A9 99		LDAIM \$99	
4380: 8E3F 8D 81 1A		STA PADD	= 10011001
4390: 8E42 A9 03		LDAIM \$03	SCAN TOETSRIJ 7-C
4400: 8E44 8D 82 1A		STA PBD	= 00000011
4410: 8E47 A9 08		LDAIM \$08	ZET 'SCHUIFBIT' IN DIREC
4420: 8E49 85 0D		STA DIREC	
4430: 8E4B AD 80 1A		LDA PAD	
4440: 8E4E 49 FF		EORIM \$FF	INVERTEER
4450: 8E50 29 66		ANDIM \$66	TOETS INGEDRUKT ?
4460: 8E52 D0 06		BNE PUSH	(7-8-B-C)
4470: 8E54 A9 00		LDAIM \$00	MAAK DIREC LEEG
4480: 8E56 85 0D		STA DIREC	

=====
 IFATE 65XX ASSEMBLER V1.0
 With permission of Fa. Proton
 lat Naarden we can offer you
 the Format Lister/Assembler/
 Tape utilities/Editor FATE.
 IFATE is a 12K program, the
 assembly listing shows like
 AIM65 listings. With the edi
 tor one is able to develop
 textfiles and sourcefiles.
 One can write the files on
 tape and read it from tape.
 With the format lister one
 is able to make hardcopies of
 the textfiles on printer.lti
 is possible to cut your ori
 ter OKI Microline 80 into sev
 eral modes. With the assem
 bler you may convert source
 files into object-code. from
 cassette to cassette. from
 memory to memory, from cas
 sette to memory, from memory
 to cassette. Merge and split
 also possible.
 IFATE is originally used as a
 Proton DOS for JUNIOR. also
 known as SENIOR. Our member
 Rob Banen catched and made a
 cassette version. He also
 developed a complete source
 listing with comments in En
 glish.
 You need a JUNIOR. two cas
 sette recorders (one is also
 possible) with motor-control
 . the Elekterminal or other
 device. OKI 80 printer or
 other printer, at least 16 K
 RAM better 32 K (from \$2000).
 IFATE is available as follows:
 - 12K on tape with manual in
 Dutch language. Fl.47.50
 - complete source-listing in
 English language. Fl.110,-
 To be paid at postal account
 184133 of Mr. W.L. van Peit
 lat Krimpen a.d. IJssel. or
 to AMRO-bank at Krimpen a.d. I
 Jssel. acc.nr.44.11.06.471.
 Foreign countries Fl. 7.50
 extra transfers.
 =====
 ASM 65C 65CXXX-CROSS-ASSEM
 BLER FOR SENIOR MONITOR
 =====
 The ASM 65C 65CXXX-CROSS-
 ASSEMBLER handles all new in
 structions of 65C02-process
 ors and is fully compatible
 with 6502. Author: Rob Banen
 Schiedam. The Netherlands.
 Complete FATE-source-listing
 of 57 pages (English).
 - Tape \$3000-\$486A Fl. 15.00
 - Source listing Fl. 52.50
 =====

4490: BE58 18		CLC	CARRY =0	
4500: BE59 60		RTS	EN RETURN	
4510: BE5A 06 OD	PUSH	ASL DIREC	SCHUIF BIT	
4520: BE5C 29 62		ANDIM \$62	TOETS B GEDRUKT ?	
4530: BE5E D0 05		BNE PUSHB		
4540: BE60 A2 ED		LDXIM \$ED	NAAR RECHTS	
4550: BE62 4C 7F 8E		JMP PUSHED		
4560: BE65 06 OD	PUSHA	ASL DIREC		
4570: BE67 29 22		ANDIM \$22	TOETS 7 GEDRUKT	
4580: BE69 D0 05		BNE PUSHB		
4590: BE6B A2 EE		LDXIM \$EE	NAAR ONDER	
4600: BE6D 4C 7F 8E	PUSHB	JMP PUSHED		
4610: BE70 06 OD		ASL DIREC		
4620: BE72 29 02		ANDIM \$02	TOETS 8 GEDRUKT ?	
4630: BE74 D0 05		BNE PUSHC		
4640: BE76 A2 EF		LDXIM \$EF	NAAR LINKS	
4650: BE78 4C 7F 8E		JMP PUSHED		
4660: BE7B 06 OD	PUSHC	ASL DIREC		
4670: BE7D A2 EC		LDXIM \$EC	NAAR BOVEN	
4680: BE7F 86 0E	PUSHED	STX DIRPM	PIJLTJE IN GREEDY	
4690: BE81 38		SEC	CARRY =1	
4700: BE82 60		RTS		
4710:				
4720:		MOVE :	VERPLAATST HCOOR NAAR	
4730:			GELANG DE GEWENSTE RICHTING	
4740:				
4750: BE83 A5 52	MOVE	LDA DIRECA	HAAL RICHTING	
4760: BE85 4A		LSRA	4 X SCHUIVEN	
4770: BE86 4A		LSRA		
4780: BE87 4A		LSRA		
4790: BE88 4A		LSRA		
4800: BE89 F0 07		BEQ	MOVEND NIETS DOEN	
4810: BE8B 4A		LSRA		
4820: BE8C D0 05		BNE NMOVRI	NAAR RECHTS ?	
4830: BE8E E6 20		INC HCOORL		
4840: BE90 F0 OD		BEQ MOVDO	CARRY ?	
4850: BE92 60	MOVEND	RTS		
4860: BE93 4A	NMOVRI	LSRA	NAAR BENEDEN ?	
4870: BE94 D0 0C		BNE NMOVDO		
4880: BE96 A5 20		LDA HCOORL	TEL ER 40 BIJ	
4890: BE98 18		CLC		
4900: BE99 69 40		ADCIM \$40		
4910: BE9B 85 20		STA HCOORL		
4920: BE9D 90 F3		BCC MOVEND	GEEN CARRY ?	
4930: BE9F E6 21	MOVDO	INC HCOORL	+01	
4940: BEA1 60		RTS		
4950: BEA2 4A	NMOVDO	LSRA		
4960: BEA3 D0 09		BNE NMOVLI	NAAR LINKS ?	
4970: BEA5 C6 20		DEC HCOORL	1 STAP TERUG	
4980: BEA7 A5 20		LDA HCOORL		
4990: BEA9 C9 FF		CMPIM \$FF	BORROW ?	
5000: BEAB F0 0A		BEQ MOVUP		
5010: BEAD 60		RTS		
5020: BEAE A5 20	NMOVLI	LDA HCOORL	NAAR BOVEN	
5030: BEBO 38		SEC	TREK ER 40 VANAF	
5040: BEB1 E9 40		SBCIM \$40		
5050: BEB3 85 20		STA HCOORL		
5060: BEB5 B0 DB		BCS MOVEND	BORROW ?	
5070: BEB7 C6 21	MOVUP	DEC HCOORL	+01	
5080: BEB9 60		RTS		
5090:				
5100:		TUNCHK :	KIJK OF TE VERPLAATSEN OBJECT	
5110:			ZICH AAN HET EINDE VAN EEN	
5120:			TUNNEL BEVINDT	

=====
 IB1JEENKOMSTEN REGIO DEN HAAG!
 De regio Den Haag/Rotterdam/
 Leiden en omstreken organi-
 seert weer haar gebruikelij-
 ke bijeenkomsten voor leden
 van en belangstellenden voor
 onze club. en wel oo :
 Zaterdag 13 april 1985
 Zaterdag 8 juni 1985
 in het Klubhuis De Boskant
 Uilebomen 71
 2511 VR Den Haag
 Tel.: 070 - 468859
 (omgeving Schenk-
 viadukt)
 organisatie: G. van Roekel.

=====
 IHULP NODIG VOOR REDAKTIEWERK!
 De redactie zoekt iemand die
 ter afwisseling in de hobby!
 bereid en in staat is goede
 vertalingen te maken van het
 Nederlands in het Engels van
 handleidingen, programma's,
 artikelen welke zijn oeu-
 biceerd of anderszins. Deze
 vertalingen worden ooggenomen
 in de oaderware-service en
 kunnen buitenlandse leden en
 belangstellenden van dienst
 zijn. Er staan geen vergoe-
 dingen voor dit werk, daar-
 voor is de club niet rijk
 genoeg. maar misschien vindt
 u het wel interessant? Bel
 u eens oo: 01807 - 19881 of
 schrijf eens naar Willem L.
 Ivan Pelt van de redactie.

=====
 fio-FORTH Source for SENIOR!
 De Elektuur JUNIOR computer
 die draait met het disk ooe-
 rating system van de firma
 Proton te Naarden wordt ook
 wel SENIOR genoemd.
 Voor deze SENIOR tyote Gert
 Ivan Oobroek de gehele source
 van de fio-FORTH in aan de
 hand van de officiele docu-
 mentatie (verkrijgbaar bij
 de redactie, vraag prijsover-
 zicht).
 Deze gehele assembly source
 listing is verkrijgbaar voor
 Fl. 48,50, incl. verpakking
 en verzendkosten.
 De object ervan is ook ver-
 krijgbaar. De object beslaat
 het opeugengebied van \$2000
 It/a \$4EB2. Kosten van de cas-
 sette: Fl. 15,00. Formaat:
 1K1M/JUNIOR Hypertade.
 Giro 841433 t.n.v. W.L. van
 Pelt te Krimozen a.d. IJssel.

PMDL2 MICRO-WARE ASSEMBLER 65XX-1.1

PAGE 09

```

5130:
5140: BEBA A2 00      TUNCHK LDXIM $00    KIJK OF HCOOR BUITEN
5150: BEBC A5 21      TUNA   LDA   HCOORL +01 HET BORD IS
5160: BEBE DD DB 8E    CMPAX TUNAA
5170: BEC1 D0 07      BNE   TUNB
5180: BEC3 A5 20      LDA   HCOORL
5190: BEC5 DD DF 8E    CMPAX TUNTAB
5200: BEC8 F0 06      BEQ   TUNC
5210: BECA E8          TUNB   INX
5220: BECB E0 04      CPXIM $04
5230: BECD D0 ED      BNE   TUNA
5240: BECF 60          RTS
5250: BED0 BD E3 8E    TUNC   LDAAX TUNTAC VERPLAATS NAAR ANDERE KANT
5260: BED3 85 21      STA   HCOORL +01
5270: BED5 BD E7 8E    LDAAX TUNTAD
5280: BED8 85 20      STA   HCOORL
5290: BEDA 60          RTS

5300:
5310: BEDB CF          TUNAA = $CF
5320: BEDC D6          = $D6
5330: BEDD D3          = $D3
5340: BEDE D3          = $D3
5350: BEDF DA          TUNTAB = $DA
5360: BEE0 DA          = $DA
5370: BEE1 3F          = $3F
5380: BEE2 75          = $75
5390: BEE3 D6          TUNTAC = $D6
5400: BEE4 D0          = $D0
5410: BEE5 D3          = $D3
5420: BEE6 D3          = $D3
5430: BEE7 9A          TUNTAD = $9A
5440: BEE8 5A          = $5A
5450: BEE9 74          = $74
5460: BEEA 40          = $40
5470:
5480:                 RESPLU : ZET OPTELBUFFER OP 0000
5490:
5500: BEEB A2 00      RESPLU LDXIM $00
5510: BEED A9 00      RESPL  LDAIM $00    BUFFER HEEFT 4 DIGITS
5520: BEEF 95 30      STAAX PLUSB
5530: BEF1 E8          INX
5540: BEF2 E0 04      CPXIM $04
5550: BEF4 D0 F7      BNE   RESPL
5560: BEF6 60          RTS

5570:
5580:                 ADD : TELT OPTELBUFFER BIJ SCORE OP
5590:
5600: BEF7 18          ADD   CLC     CARRY = 0
5610: BEF8 A2 06      LDXIM $06    SCORE 6 DIGITS
5620: BEEA A0 04      LDYIM $04    ADD 4 DIGITS
5630: BEFC CA          ADDA  DEX
5640: BEFD 88          DEY
5650: BEEF 30 15      BMI   ADDC
5660: BF00 B9 30 00    LDAAY PLUSB HAAL SCORE
5670: BF03 29 0F      ANDIM $0F
5680: BF05 7D F8 D2    ADCAX SCORD TEL OP
5690: BF08 C9 3A      CMPIM $3A  CARRY ?
5700: BF0A 30 03      BMI   ADDB
5710: BF0C 38          SEC
5720: BF0D E9 0A      SBCIM $0A  TREK ER 10 VANAF
5730: BF0F 9D F8 D2    ADDB  STAAX SCORD C BLIJFT 1
5740: BF12 4C FC 8E    JMP   ADDA  ZET IN SCORE
5750: BF15 90 13      BCC   ADDD  GEEN CARRY NAAR 10000 EN 100000
5760: BF17 FE F8 D2    INCAX SCORD

```

PMDL2 MICRO-WARE ASSEMBLER 65XX-1.1

PAGE 10

5770:	8F1A BD F8 D2	LDAAX SCORD	
5780:	8F1D C9 3A	CMPIM \$3A	CARRY ?
5790:	8F1F 30 09	BMI ADDD	
5800:	8F21 38	SEC	TREK ER 10 VANAF
5810:	8F22 E9 0A	SBCIM \$0A	C BLIJFT 1
5820:	8F24 9D F8 D2	STAAX SCORD	
5830:	8F27 CA	DEX	GA NAAR 100000
5840:	8F28 10 EB	BPL ADDC	
5850:	8F2A 20 59 8F	JSR EXHAP	
5860:	8F2D 60	RTS	
5870:			
5880:		DEPOIN :	VERLAAGT AANTAL PUNTJES MET 1
5890:			EN KIJKT OF ZE WEG ZIJN
5900:			
5910:	8F2E A2 02	DEPOIN LDXIM \$02	BEGIN ACHTERAAN
5920:	8F30 DE 7B D5	DEPA DECAX PUNTD	VERLAAG MET 1
5930:	8F33 BD 7B D5	LDAAX PUNTD	CONTROLE VOOR BORROW
5940:	8F36 C9 2F	CMPIM \$2F	
5950:	8F38 D0 08	BNE DEPB	EXIT
5960:	8F3A A9 39	LDAIM \$39	
5970:	8F3C 9D 7B D5	STAAX PUNTD	HERSTEL
5980:	8F3F CA	DEX	GA 1 DIGIT NAAR VOREN
5990:	8F40 10 EE	BPL DEPA	
6000:	8F42 A2 02	DEPB LDXIM \$02	KIJK OF ZE ALLEMAAL 0 ZIJN
6010:	8F44 BD 7B D5	DEPC LDAAX PUNTD	
6020:	8F47 C9 30	CMPIM \$30	IS HET 0 ?
6030:	8F49 D0 05	BNE DEPD	
6040:	8F4B CA	DEX	PAK VOLGENDE
6050:	8F4C 10 F6	BPL DEPC	
6060:	8F4E 38	SEC	ALLEMAAL 0 -> C =1
6070:	8F4F 60	RTS	
6080:	8F50 18	DEPD CLC	NOG NIET 0 -> C=0
6090:	8F51 60	RTS	
6100:			
6110:		CURSET :	ZET CURSOR OP ETEND MONSTER
6120:			
6130:	8F52 A9 20	CURSET LDAIM \$20	VERWIJDER GREEDY
6140:	8F54 A0 00	LDYIM \$00	
6150:	8F56 91 24	STAIY PMPOS1	
6160:	8F58 60	RTS	
6170:			
6180:		EXHAP :	KIJKT OF ER EEN EXTRA VERDIEND IS
6190:			
6200:	8F59 A6 50	EXHAP LDX EXTRA	
6210:	8F5B AD F9 D2	LDA SCORD +01 PAK 10.000 TALLEN	
6220:	8F5E DD 6B 8F	CMPAX LIM VERGELIJK MET LIMIET	
6230:	8F61 D0 07	BNE NOT	
6240:	8F63 E6 50	INC EXTRA PAK VOLGENE LIMIET	
6250:	8F65 E6 08	INC PMAANT	
6260:	8F67 20 F9 8A	JSR PMATTK TEKEN ZE	
6270:	8F6A 60	NOT RTS	
6280:			
6290:	8F6B 32	LIM = \$32	1e : 20.000
6300:	8F6C 35	= \$35	2e : 50.000
6310:	8F6D 30	= \$30	3e : 100.000
6320:	8F6E 37	= \$37	4e : 170.000
6330:	8F6F 35	= \$35	5e : 250.000
6340:	8F70 3C	= \$3C	6e : NIET TE KRIJGEN
6350:			
6360:			
6370:		BONBER :	TELT ZOVEEL KEER ALS IN A DECIMAAL STAAT
6380:			DE STERETJESBONUS BIJ EN LAAT DIT ZIEN.
6390:			
6400:	8F71 48	BONBER PHA	SAVE ACCU

6410: 8F72 4A LSRA HAAL TIENTALLEN OP
 6420: 8F73 4A LSRA
 6430: 8F74 4A LSRA
 6440: 8F75 4A LSRA
 6450: 8F76 09 30 ORAIM \$30 MAAK ER ASCII VAN
 6460: 8F78 BD F9 D3 STA TEXTD +01
 6470: 8F7B 29 0F ANDIM \$0F
 6480: 8F7D AA TAX
 6490: 8F7E A9 00 LDAIM \$00 ZET CODE OM IN HEX
 6500: 8F80 CA BONA DEX
 6510: 8F81 30 06 BMI BONB
 6520: 8F83 18 CLC
 6530: 8F84 69 0A ADCIM \$0A
 6540: 8F86 4C 80 BF JMP BONA
 6550: 8F89 85 53 BONB STA CTRLA BEWAAR EVEN
 6560: 8F8B 68 PLA
 6570: 8F8C 29 0F ANDIM \$0F EENHEDEN
 6580: 8F8E 09 30 ORAIM \$30 MAAK ASCII
 6590: 8F90 BD FA D3 STA TEXTD +02
 6600: 8F93 29 0F ANDIM \$0F
 6610: 8F95 18 CLC
 6620: 8F96 65 53 ADC CTRLA DEC - HEX IS OMGEZET
 6630: 8F98 85 53 STA CTRLA BEWAAR EVEN
 6640: 8F9A 20 30 8B JSR TXTSET ZET OP VOOR TEKST
 6650: 8F9D 20 65 89 JSR PRINT
 6660: 8FA0 09 = \$09
 6670: 8FA1 04 = \$04
 6680: 8FA2 58 = 'X
 6690: 8FA3 09 = \$09
 6700: 8FA4 7B = \$7B
 6710: 8FA5 42 = 'B
 6720: 8FA6 4F = '0
 6730: 8FA7 4E = 'N
 6740: 8FA8 55 = 'U
 6750: 8FA9 53 = 'S
 6760: 8FAA 03 = \$03
 6770: 8FAB A0 03 BONC LDYIM \$03
 6780: 8FAD B9 34 00 BOND LDAAY STARPT STERPUNTEN IN BUFFER
 6790: 8FB0 99 30 00 BOND STAAY PLUSB
 6800: 8FB3 88 DEY
 6810: 8FB4 10 F7 BPL BOND
 6820: 8FB6 20 F7 8E JSR ADD
 6830: 8FB9 20 58 8B JSR DELAYA
 6840: 8FBC C6 53 DEC CTRLA
 6850: 8FBE D0 EB BNE BONC
 6860: 8FC0 20 63 8B JSR TXTSOW LAAT TEKST EVEN ZIEN
 6870: 8FC3 60 RTS
 6880:
 6890: COMPAR : VERGELIJKT SCORE MET HISCORE IN
 6900: TABEL, AANGEWEZEN DOOR VELDP
 6910:
 6920: 8FC4 A0 00 COMPAR LDYIM \$00 VERGELIJK HI-SCORES
 6930: 8FC6 B9 F8 D2 HCOMP LDAAY SCORD
 6940: 8FC9 D1 02 CMPY VELDPL
 6950: 8FCB 30 07 BMI LOWER NIET DE HOOGSTE
 6960: 8FCD D0 07 BNE HIGHER GELIJK ?
 6970: 8FCF C8 INY PAK VOLGEND GETAL
 6980: 8FD0 C0 06 CPYIM \$06 ALLES GEHAD ?
 6990: 8FD2 D0 F2 BNE HCOMP
 7000: 8FD4 18 LOWER CLC C=0, SCORE LAGER
 7010: 8FD5 60 RTS
 7020: 8FD6 38 HIGHER SEC C=1, SCORE HOGER
 7030: 8FD7 60 RTS
 7040:
 7050: TURN : DRAAIT 'RICHTING' IN A OM
 7060:
 7070: 8FD8 48 TURN PHA SAVE ACCU
 7080: 8FD9 29 50 ANDIM \$50 LINKS/RECHTS ?
 7090: 8FDB F0 04 BEQ BOVON
 7100: 8FDD 68 PLA
 7110: 8FDE 49 50 EORIM \$50 OMWISSELEN
 7120: 8FE0 60 RTS
 7130: 8FE1 68 BOVON PLA
 7140: 8FE2 49 A0 EORIM \$A0 BOVEN/BENEDEN
 7150: 8FE4 60 RTS
 7160:
 7170: einde deel 2
 7180:
 7190: 00 90 NMWRT * \$9000

```
19 22 QLIST
SCR # 19
0 ( Utilities.    WAIT.    BORDER.    SCR-WRITE      )
1 ( naar Willem R. Zeeff      )
2 FORTH DEFINITIONS HEX
3 : WAIT 0 DO 33E 0 DO LOOP LOOP : ( delav      )
4 : BORDER OF AND 400 *
5       6000 ( top of FORTH's memory assionment      )
6       ( installation dependent !      )
7       + ;
8
9 : SCR-WRITE DUP 1A79 C! ( tape id number      )
10    BORDER DUP 1A70 ! ( start adress      )
11    400 + 1A72 ! ( end adress      )
12    WRITE :
13
14 --)
15

SCR # 20
0 ( Utilities.    SCR-READ.    TO-TAPE.    FROM-TAPE      )
1 ( naar Willem R. Zeeff      )
2 : SCR-READ 1A79 C! ( tape id number      )
3     READ :
4 : TO-TAPE 1+ SWAP 27 WAIT ( delav )
5     DO CR OA WAIT ( delav )
6     ." WRITING" I 4 .R ( screennumber      )
7     SPACE I SCR-WRITE
8     ." READY "
9     LOOP :
10 : FROM-TAPE 1+ SWAP DO CR
11     ." READING" I 4 .R ( screennumber      )
12     SPACE I SCR-READ ." READY "
13     LOOP :
14
15 --)

SCR # 21
0 ( disk utilities          Gert Klein      )
1
2
3 : DISK FLUSH 0 I/OFLAG ! : ( select disk as i/o )
4 : TAPE FLUSH 1 I/OFLAG ! : ( select tape as i/o )
5
6 : INBUF B/SCR * OFFSET @ ( offset to other drives )
7     + SWAP B/SCR * ( source block number      )
8     B/SCR OVER + ( taroet block number      )
9     SWAP DO DUP
10    I BLOCK ( load source block      )
11    2 - ! ( assion taroet number      )
12    1+ UPDATE ( uodate buffer      )
13    LOOP DROP :
14
15 --)

SCR # 22
0 ( disk utilities          Gert Klein      )
1
2 : T->D ( copy from tape to disk      )
3     TAPE INBUF 0 I/OFLAG ! ( select i/o      )
4     FLUSH :
5
6 : D->T ( copy from disk to tape      )
7     DISK INBUF 1 I/OFLAG ! ( select i/o      )
8     FLUSH DISK :
9
10 DECIMAL
11 LATEST 12 +ORIGIN ! ( lock into svstem      )
12 HERE 28 +ORIGIN !
13 HERE 30 +ORIGIN !
14 HERE FENCE !
15 ;S
OK
```

>PR

0010 : INFORMATION ABOUT OSI NINE DIGIT BASIC

0020 :

0030 :

0040 : COLLECTED BY: GERT KLEIN
0050 : DIEDENWEG 119
0060 : 6706 CM WAGENINGEN

0070 :

0080 : BASED ON A PREVIOUS PUBLICATION OF WILLEM VAN PEEL FOR THE
0090 : KB9 BASIC FOR THE JUNIOR. THIS VERSION APPLIES TO THE
0100 : BASIC OF THE OHIO SCIENTIFIC OS-65D V3.3 DISK OPERATING
0110 : SYSTEM. PLEASE LET US KNOW IF YOU FIND ANY INCORRECTNESSES
0120 : IN THIS LISTING.

0130 :

0140 :

0150	. BA \$00	
0160	NLJMP . DE \$00	: NEW LINE JUMP
0170	Y/AJMP . DE \$03	: JUMP TO PRINT STRING ROUTINE
0180	VECFIXTFLO . DE \$06	: VECTOR TO 'FIXED-TO-FLOATING' SUB
0190	VECFLOTFIX . DE \$08	: VECTOR TO 'FLOATING-TO-FIXED' SUB
0200	SRCHCHAR . DE \$0A	: SEARCH CHARACTER
0210	SCANBTWNQW . DE \$0B	: SCAN-BETWEEN-QUOTES FLAG
0220	INPBPNTR . DE \$0C	: INPUT BUFFER POINTER: # OF SUBSCR.
0230	DEFDIMFLG . DE \$0D	: DEFAULT DIM FLAG
0240	TYPE1 . DE \$0E	: TYPE: FF=STRING. 00=NUMERIC
0250	TYPE2 . DE \$0F	: TYPE: 80=INTEGER. 00=FLOATING P.
0260	FLAG . DE \$10	: FLAG: DATA SCAN: LIST QUOTE: MEMORY
0270	SUBSCRFNX . DE \$11	: SUBSCRIPT FLAG: FNX FLAG
0280	SIGN . DE \$12	: \$00=INPUT. \$98=READ
0290	CMPEVALFLG . DE \$13	: COMPARISON EVALUATION FLAG
0300	INPFLG . DE \$14	: INPUT FLAG (SUPPRESS OUTPUT)
0310	POSPRLN . DE \$16	: POSITION ON PRINT LINE
0320	PRLNWIDTH . DE \$17	: MAXIMUM PRINT LINE WIDTH
0330	COLLIMIT . DE \$18	: INPUT COLUMN LIMIT
0340	INTEGER . DE \$19	: INTEGER VALUE (FOR GOTO ETC)
0350	BUFFER . DE \$1B	: START OF INPUT BUFFER
0360	:	
0370	. BA \$62	
0380	BUFFEREND . DE \$62	: END OF INPUT BUFFER
0390	DESCRSTCK . DE \$63	: POINTERS FOR DESCRIPTOR STACK
0400	TEMPSTRNG . DE \$66	: DESCRIPTOR STACK (TEMP STRING)
0410	STCKEND . DE \$6E	: END OF DESCRIPTOR STACK
0420	UTILPNTR . DE \$6F	: UTILITY POINTER AREA
0430	MULTPROD . DE \$73	: PRODUCT AREA FOR MULTIPLICATION
0440	STRTBAS . DE \$78	: POINTER: START-OF-BASIC
0450	STRTVAR . DE \$7A	: POINTER: START-OF-VARIABLES
0460	STRTARR . DE \$7C	: POINTER: START-OF-ARRAYS
0470	ENDARR . DE \$7E	: POINTER: END-OF-ARRAYS
0480	STRNGSTOR . DE \$80	: STRING-STORAGE (MOVING DOWN)
0490	STRNGPNTR . DE \$82	: UTILITY STRING POINTER
0500	MEMEND . DE \$84	: POINTER: LIMIT-OF-MEMORY
0510	LINENM . DE \$86	: CURRENT BASIC LINE NUMBER
0520	PREVLINENM . DE \$88	: PREVIOUS BASIC LINE NUMBER
0530	STATCONT . DE \$8A	: POINTER: BASIC STATEMENT FOR CONT
0540	DATALINENM . DE \$8C	: CURRENT DATA LINE NUMBER
0550	DATAADDRS . DE \$8E	: CURRENT DATA ADDRESS
0560	INPVEC . DE \$90	: INPUT VECTOR
0570	CURVARNAM . DE \$92	: CURRENT VARIABLE NAME
0580	CURVARADDR . DE \$94	: CURRENT VARIABLE ADDRESS
0590	FNVARPNTR . DE \$96	: VARIABLE POINTER FOR FOR/NEXT
0600	STRTRWKARE . DE \$98	: START OF WORK AREA. POINTERS. ETC.
0610	VECFUNCT . DE \$A1	: JUMP VECTOR FOR FUNCTIONS
0620	MISCNUMARE . DE \$A4	: MISC NUMERIC WORK AREA
0630	ACCU1 . DE \$A9	
0640	ACCU2 . DE \$AB	
0650	ACCUM1EXP . DE \$AE	: ACCUM#1: EXPONENT
0660	ACCUM1MANT . DE \$AF	: ACCUM#1: MANTISSA
0670	ACCUM1SIGN . DE \$B3	: ACCUM#1: SIGN
0680	CONSTPNT . DE \$B4	: SERIES EVALUATION CONSTANT POINT
0690	ACCUM1HO . DE \$B5	: ACCUM#1: HI-ORDER (OVERFLOW)

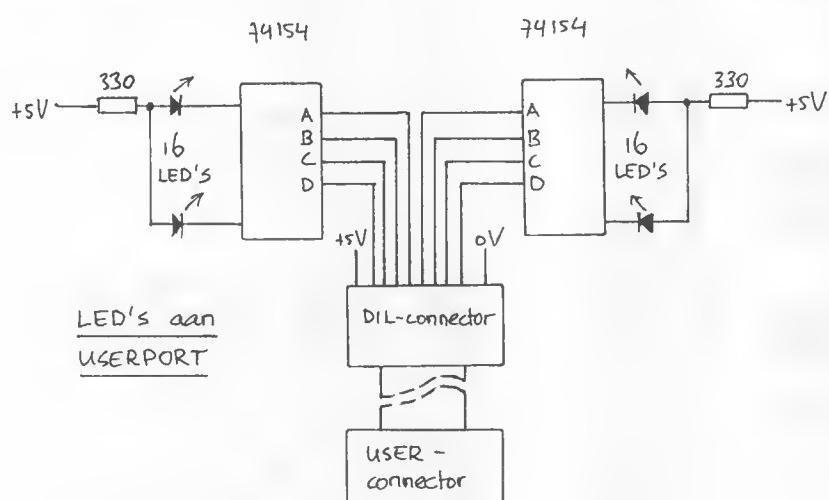
0700	ACCUM2EXP	. DE \$B6	: ACCUM#2: EXPONENT, ETC.
0710	SGNCOMP	. DE \$BC	: SIGN COMPARISON. ACCU#1 VS #2
0720	ROUNDING	. DE \$BD	: ACCU#1: LO-ORDER (ROUNDING)
0730	SERPNTR	. DE \$BE	: SERIES POINTER
0740	NEWCHA	. DE \$C0	: CHRGET SUBR.: GET BASIC CHAR
0750	PREVCHAR	. DE \$C6	: SUB ENTRY: GET PREV CHAR
0760	BUPTR	. DE \$C7	: BASIC POINTER (WITHIN SUBRTN)
0770		.	
0780		.	
0790		.	
0800		.	
0810		:	SOME ROUTINES FROM OSI BASIC
0820		.	
0830	PRIMKEYW	. DE \$0200	: ACTION ADDRS FOR PRIM KEYWORDS
0840	FUNCADDR	. DE \$0238	: ACTION ADDRS FOR FUNCTIONS
0850	OPERADDR	. DE \$0266	: HIERARCHY & ACT. ADDRS FOR OPERATORS
0860	KEYWTAB	. DE \$0284	: TABLE OF BASIC KEYWORDS
0870	MESSTAB	. DE \$0364	: BASIC MESSAGES. MOSTLY ERROR MESS.
0880	STCKSRCH	. DE \$03A1	: SEARCH STACK FOR -FOR- OR -GOSUB ACTIV.
0890	OPSPCMEM	. DE \$03CF	: OPEN UP SPACE IN MEMORY
0900	STCKTDP	. DE \$0412	: TEST: STACK TOO DEEP ?
0910	CHKAVMEM	. DE \$041F	: CHECK AVAILABLE MEMORY
0920	FATAL	. DE \$044C	: SEND CANNED ERROR MESS.. THEN:
0930	ERROR	. DE \$044E	: PRINT 'ERROR'
0940	RETWRIT	. DE \$0474	: WARM START: WAIT FOR BASIC COMMAND
0950	NWLINE	. DE \$0496	: HANDLE NEW BASIC LINE INPUT
0960	RECOMP	. DE \$051D	: REBUILD CHAINING OF BASIC LINES
0970	GETLINE	. DE \$0558	: RECEIVE LINE FROM KEYBOARD
0980	BEGINP	. DE \$0559	
0990	ENCODE	. DE \$05A7	: CRUNCH KEYWORDS INTO BASIC TOKENS
1000	RECHERCHE	. DE \$0633	: SEARCH BASIC FOR GIVEN LINE NUMBER
1010	NEW	. DE \$0662	: PERFORM NEW
1020	CLEAR	. DE \$067C	: PERFORM CLEAR
1030	RESET	. DE \$0695	: RESET THE STACK
1040	GETPTR	. DE \$06AB	: RESET BASIC EXECUTION TO START
1050	LIST	. DE \$06B9	: PERFORM LIST
1060	FOR	. DE \$0748	: PERFORM FOR
1070	EXECNXT	. DE \$07B4	: EXECUTE NEXT STATEMENT
1080	EXECCUR	. DE \$07E0	: EXECUTE BASIC STATEMENT
1090	EXECTOK	. DE \$07E3	: EXECUTE STATEMENT TOKEN
1100	RESTORE	. DE \$080A	: PERFORM RESTORE
1110	CHKSTOP	. DE \$0819	: CHECK STOP KEY
1120	STOPEND	. DE \$0828	: PERFORM STOP
1130	END	. DE \$082A	: PERFORM END
1140	ENDSTP	. DE \$082D	
1150	CONT	. DE \$0853	: PERFORM CONT
1160	RUN	. DE \$087E	: PERFORM RUN
1170	GOSUB	. DE \$0889	: PERFORM GOSUB
1180	GOTO	. DE \$08A6	: PERFORM GOTO
1190	RETURN	. DE \$08D3	: PERFORM RETURN. THEN:
1200	DATA	. DE \$08F9	: PERFORM DATA: SKIP STATEMENT
1210	NXTSTM	. DE \$0907	: SCAN FOR NEXT BASIC STATEMENT
1220	LINESEP	. DE \$090A	: SCAN FOR NEXT BASIC LINE
1230	IF	. DE \$0929	: PERFORM IF. AND PERHAPS:
1240	REM	. DE \$093C	: PERFORM REM: SKIP LINE
1250	ON	. DE \$094C	: PERFORM ON
1260	COLLECT	. DE \$096C	: INPUT FIXED-POINT NUMBER
1270	LET	. DE \$09A6	: PERFORM LET
1280	CRLF1	. DE \$0A73	: PRINT RETURN/LINE FEED
1290	PRSTRMEM	. DE \$0AC7	: PRINT STRING FROM MEMORY
1300	PRSTRYA	. DE \$0ACC	: PRINT STRING POINTED BY Y/A
1310	PRFORMCHA	. DE \$0AE9	: PRINT SINGLE FORMAT CHAR.
1320	BADDATA	. DE \$0B0D	: HANDLE BAD INPUT DATA
1330	INPUT	. DE \$0B2C	: PERFORM INPUT
1340	PRMPT	. DE \$0B4F	: PROMPT AND RECEIVE INPUT
1350	READ	. DE \$0B58	: PERFORM READ
1360	INPERR	. DE \$0C27	: CANNED INPUT ERROR MESSAGES
1370	NEXT	. DE \$0C4B	: PERFORM NEXT
1380	CHKTYP	. DE \$0CB9	: CHECK TYPE MISMATCH
1390	EVALEX	. DE \$0CCD	: EVALUATE EXPRESSION
1400	NOT	. DE \$0DEA	: PERFORM NOT
1410	GETPAR	. DE \$0E07	: EVALUATE EXPR. WITHIN PARENTH.
1420	CHKPAR	. DE \$0E0D	: CHECK PARENTHESIS. COMMA

1430	SYNTERR	. DE \$0E1E	SYNTAX ERROR EXIT
1440	FUNCSET	. DE \$0E23	SETUP FOR FUNCTIONS
1450	VARNAMESET	. DE \$0E2A	VARIABLE NAME SETUP
1460	FUNCREF	. DE \$0E4A	SETUP FUNCTION REFERENCES
1470	OR	. DE \$0E89	PERFORM OR
1480	AND	. DE \$0E8C	PERFORM AND
1490	COMP	. DE \$0E89	PERFORM COMPARISONS
1500	DIM	. DE \$0F24	PERFORM DIM
1510	VARSRCH	. DE \$0F2E	SEARCH FOR VARIABLE
1520	CHKALPH	. DE \$0FB8	
1530	NEUVAR	. DE \$0FC2	CREATE NEW VARIABLE
1540	ARRPNTR	. DE \$1028	SETUP ARRAY POINTER
1550	EVINTEXPR	. DE \$1039	EVALUATE INTEGER EXPRESSION
1560	FLPINT	. DE \$1047	FLOATING POINT TO INTEGER
1570	FNDARRAY	. DE \$1059	FIND OR MAKE ARRAY
1580	ILLEG	. DE \$10D0	MESS. ILLEGAL QUALITY ERROR
1590	FRE	. DE \$1204	PERFORM FRE, AND:
1600	INTFLP	. DE \$1218	CONVERT FIXED-TO-FLOATING
1610	POS	. DE \$1225	PERFORM POS
1620	NONDRCT	. DE \$122B	CHECK NOT DIRECT
1630	DEF	. DE \$1235	PERFORM DEF
1640	FNXSYNT	. DE \$1273	CHECK FNX SYNTAX
1650	EVFNX	. DE \$1276	EVALUATE FNX
1660	STRSTR	. DE \$12E9	PERFORM STR\$
1670	STRVEC	. DE \$12F9	DO STRING VECTOR
1680	STRSET	. DE \$130B	SCAN, SETUP STRING
1690	DESCR	. DE \$1372	BUILD DESCRIPTOR
1700	GARBCOLL	. DE \$1394	GARBAGE COLLECTION
1710	CONC	. DE \$14B7	CONCATENATE
1720	STRSTORE	. DE \$14F4	STORE STRING
1730	DSCRDSTR	. DE \$151D	DISCARD UNWANTED STRING
1740	CLNDESCR	. DE \$1555	CLEAN DESCRIPTOR STACK
1750	CHRSTR	. DE \$1566	PERFORM CHR\$
1760	LEFTSTR	. DE \$157A	PERFORM LEFT\$
1770	RIGHTSTR	. DE \$15A6	PERFORM RIGHT\$
1780	MIDSTR	. DE \$15B1	PERFORM MID\$
1790	PULLSTR	. DE \$15D9	PULL STRING DATA
1800	LEN	. DE \$15F6	PERFORM LEN
1810	STRTONUM	. DE \$15FC	SWITCH STRING TO NUMERIC
1820	ASC	. DE \$1605	PERFORM ASC
1830	GETINTX	. DE \$1615	GET BYTE PARAMETER
1840	VAL	. DE \$1627	PERFORM VAL
1850	PARSPOKE	. DE \$1666	GET 2 PARAMS FOR POKE
1860	FLPNUMB	. DE \$1672	CONVERT FLOATING-TO-FIXED
1870	PEEK	. DE \$1688	PERFORM PEEK
1880	POKE	. DE \$1693	PERFORM POKE
1890	TRAP	. DE \$169C	PERFORM TRAP
1900	ADD05	. DE \$16B8	ADD 0.5
1910	SUBTR	. DE \$16BF	PERFORM SUBTRACTION
1920	PLUS	. DE \$16D1	PERFORM ADDITION
1930	COMPLACCU1	. DE \$17E8	COMPLEMENT ACCUM #1
1940	OVEREXIT	. DE \$181F	OVERFLOW EXIT
1950	MULTBYT	. DE \$1824	MULILPLY-A-BYTE
1960	CONSTANTS	. DE \$1885	CONSTANTS
1970	LOG	. DE \$18B3	PERFORM LOG
1980	MULT	. DE \$18F1	PERFORM MULTIPLICATION
1990	UNPACK	. DE \$1987	UNPACK MEMORY INTO ACCUM #2
2000	TSTACCUMS	. DE \$19B2	TEST \$ ADJUST ACCUMULATORS
2010	OVERUNDER	. DE \$19CF	HANDLE OVERFLOW AND UNDERFLOW
2020	MULT10	. DE \$19DD	MULTIPLY BY 10
2030	TENINFL	. DE \$19F4	10 IN FLOATING BINARY
2040	DIVIDE10	. DE \$19F9	DIVIDE BY 10
2050	DIVBY	. DE \$1AOA	PERFORM DIVIDE-BY
2060	DIVINTO	. DE \$1AOF	PERFORM DIVIDE-INTO
2070	GETCNSTYA	. DE \$1A9D	UNPACK MEMORY INTO ACCUM #1
2080	PACK	. DE \$1AC2	PACK ACCUM #1 INTO MEMORY
2090	COPY	. DE \$1AF7	MOVE ACCUM #2 TO #1
2100	YPOC	. DE \$1B07	MOVE ACCUM #1 TO #2
2110	ROUND	. DE \$1B16	ROUND ACCUM #1
2120	ACCU1SGN	. DE \$1B26	GET ACCUM #1 SIGN
2130	SGN	. DE \$1B34	PERFORM SIGN
2140	ABS	. DE \$1B53	PERFORM ABS

2150	CMPFLPYA	. DE \$1B56	: COMPARE ACCUM #1 TO MEMORY
2160	FLPFINT	. DE \$1B96	: FLOATING-TO-FIXED
2170	INT	. DE \$1BC7	: PERFORM INT
2180	STRFLP	. DE \$1BEE	: CONVERT STRING TO FLP
2190	FLPPLA	. DE \$1C8D	: GET NEW ASCII DIGIT
2200	CONST2	. DE \$1CC2	: CONSTANTS
2210	PRIN	. DE \$1CD1	: PRINT IN, THEN:
2220	PRLINENUM	. DE \$1CD8	: PRINT BASIC LINE NUMBER
2230	PRNUMAX	. DE \$1CDC	
2240	PRIAX	. DE \$1CEC	: CONVERT FLP TO ASCII
2250	CONST3	. DE \$1E1C	: CONSTANTS
2260	SQR	. DE \$1E45	: PERFORM SQR
2270	POWER	. DE \$1E4F	: PERFORM POWER FUNCTION
2280	NEGATE	. DE \$1E88	: PERFORM NEGATION
2290	CONST4	. DE \$1E93	: CONSTANTS
2300	EXP	. DE \$1EC1	: PERFORM EXP
2310	SEREVAL	. DE \$1F14	: SERIES EVALUATION
2320	RNDCONST	. DE \$1F5E	: RND CONSTANTS
2330	RND	. DE \$1F66	: PERFORM RND
2340	COS	. DE \$1FA2	: PERFORM COS
2350	SIN	. DE \$1FA9	: PERFORM SIN
2360	TAN	. DE \$1FF2	: PERFORM TAN
2370	CONST5	. DE \$201E	: CONSTANTS
2380	PRINT	. DE \$2059	: PERFORM PRINT
2390	EXIT	. DE \$223C	: PERFORM EXIT
2400	DISK	. DE \$2253	: PERFORM DISK
2410	DISK!	. DE \$2259	: PERFORM DISK!"....."
2420	DISKCL	. DE \$2283	: PERFORM DISK CLOSE
2430	DISKOP	. DE \$2299	: PERFORM DISK OPEN
2440	EDIT	. DE \$3770	: LINE EDITOR
2450			
2460			
2470		. EN	

//

BBC microcomputer LED's aan USERPOORT



J. van Huffel
Van Paestraat 81
B - 1080 BRUSSEL

Aan de Tekentafel:
Fridus Jonkman

adressen : DDRB \$FE62 : 1 vooruitgang (FF)
ORB \$FE60

zet men in adres \$FE6B de waarde \$C0, en start met de timer op adres \$FE65 met bijvoorbeeld de waarde \$FF, dan knipoert de LED, die overeenstemt met PB7, onafhankelijk van de werking van de computer.

LPRINT

0010:
 0020: LARGE CHARACTERS WITH THE MICROLINE-80
 0030: By Frans Bakx
 0040: Huissteden 1112
 0050: 6605 HD Wijchen
 0060: tel. 08894 - 16389
 0070: I present here a set of subroutines.
 0080: These subroutines can be used to print
 0090: large characters: 3 lines hi and 4 normal
 0100: characters width. The point patterns to
 0110: write a character are placed in a table.
 0120: In the table below you can read how
 0130: many characters can be printed on normal
 0140: 9 inch paper.
 0150:

	short line	long line
compressed	26	33
normal	16	20
elongated	8	10

 0160:
 0170:
 0180:
 0190:
 0200:
 0210:
 0220:
 0230:
 0240:
 0250:
 0260:
 0270:
 0280:
 0290:
 0300:
 0310:
 0320:
 0330:
 0340:
 0350:
 0360:
 0370:
 0380:
 0390:
 0400:
 0410:
 0420:
 0430: 0200 LPRINT ORG \$0200
 0440:
 0450: E4 00 LCOUNT * \$00E4 counter of lines (1-4)
 0460: E5 00 NCHAR * \$00E5 number of characters to place in BUFFER
 0470: E6 00 CURAD * \$00E6 pointer to first character to place in BUFFER
 0480: E8 00 POINT * \$00E8 pointer to patterns in TABLE
 0490: 00 03 BUFFER * \$0300 string buffer
 0500: 00 04 TABLE * \$0400 table of patterns
 0510: Junior subroutines X,Y unchanged
 0520: E8 11 CRLF * \$11E8 output a cr and a lf
 0530: 34 13 PRCHA * \$1334 output a character in accu
 0540:
 0550: 0200 8A TXA
 0560: 0201 48 PHA save X and Y registers
 0570: 0202 98 TYA
 0580: 0203 48 PHA
 0590: 0204 20 78 02 JSR FILLBF fill buffer
 0600: 0207 20 0F 02 JSR PRBUFF print buffer
 0610: 020A 68 PLA
 0620: 020B A8 TAY restore X and Y
 0630: 020C 68 PLA
 0640: 020D AA TAX

LPRINT

```

0650: 020E 60          RTS      end
0660:
0670:                   subroutine PRBUFF print all the characters
0680:                   up to $00 .
0690: 020F A9 01          PRBUFF LDAIM $01   first line
0700: 0211 85 E4          STA    LCOUNT
0710: 0213 20 22 02          PRBF   JSR    PRLINE  print one line
0720: 0216 20 E8 11          JSR    CRLF
0730: 0219 E6 E4          INC    LCOUNT next line
0740: 021B A5 E4          LDA    LCOUNT
0750: 021D C9 04          CMPIM $04   maximal 4 lines
0760: 021F D0 F2          BNE    PRBF   if any
0770: 0221 60          RTS
0780:
0790:                   subroutine PRLINE print one of the 3 lines.
0800:                   The line number is in LCOUNT.
0810: 0222 A2 00          PRLINE LDIXIM $00  set up for first char.
0820: 0224 BD 00 03          PRLN  LDAAX BUFFER get one character from BUFFER
0830: 0227 F0 06          BEQ    PRLND $00 is terminator
0840: 0229 20 30 02          JSR    PRCHAR print the patterns
0850: 022C E8          INX    next character
0860: 022D 10 F5          BPL    PRLN  unconditional branch
0870: 022F 60          PRLND RTS
0880:
0890:                   subroutine PRCHAR print the 3 horizontal
0900:                   patterns of a character. The patterns are
0910:                   found in TABLE by an offset. The offset
0920:                   is related on charactercode -$20 and the
0930:                   number of the printing line (in LCOUNT).
0940: 0230 A0 00          PRCHAR LDYIM TABLE low byte of TABLEs start
0950: 0232 84 E8          STY    POINT
0960: 0234 A0 04          LDYIM TABLE /255 hi byte of TABLEs start
0970: 0236 84 E9          STY    POINT +01
0980: 0238 A0 03          LDYIM $03  set up for 4 blanks
0990: 023A 38          SEC
1000: 023B E9 20          SBCIM $20  character:=character-$20
1010: 023D 90 1E          BCC    PRCHND no character below $20
1020: 023F F0 19          BEQ    SPACES print large space (Y=3)
1030: 0241 A8          TAY    character-$20->Y reg
1040: 0242 A9 09          LDAIM $09  each character takes 9 patterns
1050: 0244 20 5E 02          JSR    PNTSUM point:=point + 9 * (Y-1)
1060: 0247 A4 E4          LDY    LCOUNT set line number
1070: 0249 A9 03          LDAIM $03  each line takes 3 patterns
1080: 024B 20 5E 02          JSR    PNTSUM point:=point + 3 * (Y-1) and return with Y=0
1090: 024E B1 E8          PRCHR  LDAIY POINT set the pattern
1100: 0250 20 34 13          JSR    PRCHA print it
1110: 0253 C8          INY    next pattern
1120: 0254 C0 03          CPYIM $03  maximal 3 bytes/character
1130: 0256 D0 F6          BNE    PRCHR
1140: 0258 A0 00          LDYIM $00  one space
1150: 025A 20 6F 02          SPACES JSR    PRSP  print one or more spaces
1160: 025D 60          PRCHND RTS
1170:
1180:                   subroutine PNTSUM point:=point + accu * (Y-1)
1190:                   set POINT to the patterns in TABLE.
1200: 025E 88          PNTSUM DEY
1210: 025F F0 0D          BEQ    SUMEND
1220: 0261 48          PHA    save accu
1230: 0262 18          CLC
1240: 0263 65 E8          ADC    POINT point:=point + accu
1250: 0265 85 E8          STA    POINT
1260: 0267 90 02          BCC    PNSM
1270: 0269 E6 E9          INC    POINT +01
1280: 026B 68          PNSM  PLA

```

LPRINT

```

1290: 026C D0 F0      BNE    PNTSUM unconditional branch
1300: 026E 60      SUMEND RTS      return with Y=$00
1310:
1320:          subroutine PRSP print at least one blank (= $80)
1330:          or a number of blanks if Y contains a positive
1340:          number.
1350: 026F A9 80      PRSP   LDAIM $80      orafic space character
1360: 0271 20 34 13      JSR    PRCHA      print it
1370: 0274 88      DEY
1380: 0275 10 F8      BPL    PRSP
1390: 0277 60      RTS
1400:
1410:          subroutine FILLBF fills the buffer with ascii
1420:          characters in the range of $20 to $7F.
1430:          The address of the first character to store in
1440:          BUFFER must be in CURAD. The number of charac-
1450:          ters to store must be in NCHAR.
1460: 0278 A0 00      FILLBF LDYIM $00
1470: 027A A2 00      LDXIM $00
1480: 027C B1 E6      FILL   LDAIY CURAD  set character
1490: 027E 30 08      BMI    FIL      no character beyond $7F
1500: 0280 C9 20      CMPIM $20
1510: 0282 30 04      BMI    FIL      no character below $20
1520: 0284 9D 00 03      STAAX BUFFER
1530: 0287 E8      INX      next character to place in BUFFER
1540: 0288 C8      INY      next character to get
1550: 0289 E4 E5      CPX    NCHAR      all character stored?
1560: 028B D0 EF      BNE    FIL
1570: 028D A9 00      LDAIM $00      mark the end of the
1580: 028F 9D 00 03      STAAX BUFFER      string with $00
1590: 0292 60      RTS
1600:

```

TEST

```

0010:          A test program to print laroe characters with the
0020:          Microline 80.
0030:          By frans bakx
0040:          The following text will be printed in all possible
0050:          large formats :
0060:          6502
0070:          Kenner
0080:
0090: 2000      TEST    ORG    $2000
0100:
0110: 5F 10 LABJUN *      $105F      exit to Juniors monitor
0120: E8 11 CRLF *      $11E8      output a cr and lf
0130: 34 13 PRCHA *      $1334      output the character in accu
0140: 00 02 LPRINT *      $0200      print large characters.
0150:          the number of characters in NCHAR.
0160:          the startaddress of the string in CURAD
0170: E5 00 NCHAR *      $00E5
0180: E6 00 CURAD *      $00E6
0190:
0200: 2000 20 0F 20      JSR    SIXLPI      set the printer: 6 lines / inch
0210: 2003 20 33 20      JSR    FORMAT      print the text in 3 formats
0220: 2006 20 17 20      JSR    EIGHTL      set the printer: 8 lines / inch
0230: 2009 20 33 20      JSR    FORMAT
0240: 200C 4C 5F 10      JMP    LABJUN      exit to monitor
0250:
0260: 200F 20 1F 20      SIXLPI JSR    ESC      print esc ($1B)
0270: 2012 A9 36      LDAIM '6      print 6
0280: 2014 4C 34 13      JMP    PRCHA      and return
0290:
0300: 2017 20 1F 20      EIGHTL JSR    ESC
0310: 201A A9 38      LDAIM '8      print 8
0320: 201C 4C 34 13      JMP    PRCHA      and return
0330:
0340: 201F A9 1B      ESC    LDAIM $1B      output esc
0350: 2021 4C 34 13      JMP    PRCHA      and return

```

DE 6502 KENNER

```

0360:
0370: 2024 A9 1D      CMPRSD LDAIM $1D    output GS ( print compressed characters)
0380: 2026 4C 34 13    JMP     PRCHA   and return
0390:
0400: 2029 A9 1E      NORMAL LDAIM $1E    output RS ( print normal characters)
0410: 202B 4C 34 13    JMP     PRCHA   and return
0420:
0430: 202E A9 1F      ELNGTD LDAIM $1F    output US (print elongated characters)
0440: 2030 4C 34 13    JMP     PRCHA   and return
0450:
0460: 2033 20 24 20    FORMAT JSR      CMPRSD print compressed
0470: 2036 20 48 20    JSR      TEXT    print the text
0480: 2039 20 29 20    JSR      NORMAL   print normal format
0490: 203C 20 48 20    JSR      TEXT
0500: 203F 20 2E 20    JSR      ELNGTD print elongated format
0510: 2042 20 48 20    JSR      TEXT
0520: 2045 4C E8 11    JMP     CRLF   output cr and lf and return
0530:
0540: 2048 A2 05      TEXT   LDXIM $05
0550: 204A BD 62 20    LDAAX TABLE  set number of characters
0560: 204D 85 E5      STA    NCHAR
0570: 204F CA          DEX
0580: 2050 BD 62 20    LDAAX TABLE  set startaddress of text
0590: 2053 85 E7      STA    CURAD +01
0600: 2055 CA          DEX
0610: 2056 BD 62 20    LDAAX TABLE
0620: 2059 85 E6      STA    CURAD
0630: 205B 20 00 02    JSR    LPRINT print the text in large characters
0640: 205E CA          DEX
0650: 205F 10 E9      BPL    TEXT   +02
0660: 2061 60          RTS
0670:
0680: 2062 6C          TABLE  =      TXT    start of string 2 low
0690: 2063 20          =      TXT    /255 start of string 2 hi
0700: 2064 06          =      $06   length of string 2
0710: 2065 68          =      TXTS  start of string 1 low
0720: 2066 20          =      TXTS  /255 start of string 1 hi
0730: 2067 04          =      $04   length of string 1
0740:
0750: 2068 36          TXTS  =      '6
0760: 2069 35          =      '5
0770: 206A 30          =      '0
0780: 206B 32          =      '2
0790: 206C 4B          TXT   =      'K
0800: 206D 65          =      'e
0810: 206E 6E          =      'n
0820: 206F 6E          =      'n
0830: 2070 65          =      'e
0840: 2071 72          =      'r      next page ==> ==> ==> ==>
=====
```

DISK OPERATING SYSTEM DOSAV2.0A van KOEN VAN NIEUWENHOVE

Single en Double Density met JUNIOR op 5" floppy's

In editie 35 van december 1984 maakte onze jury bekend dat Koen van Nieuwenhove uit België met zijn DOS in aanmerking kwam voor de eerste prijs toegekend aan publikaties welke door hun omvang en geleverde inspanning, de praktische toepasbaarheid en de leesbaarheid het belang van de leden hebben oediend. Het betreft hier het DOS zoals dat gebruikt wordt op de redactie. Effektief kan er 128K op floppy worden weggeschreven op 40 tracks met 26 sectoren per track en 128 bytes per sector. IBM-compatibel dus. In het utility pakket komen o.a. Read en Save voor. Teneinde ook op de cassettes te kunnen blijven werken. Verder zijn er een aantal commando's, waarvan de belangrijkste zijn: Run, Limits, Go, Quit, Execute, Copy, Delete, Directory, Free, List (de sectoren kunnen op screen gedrukt worden in hex. in ascii of met alle tabs geexpanded en (CR) en (LF) uitgevoerd), Rename, Trace, Store, Format (met of zonder check), Random, Undelete (heel belangrijk!), Memtest, Validate, Backup (ge-selecteerd per file of volledige schijf copiëren), Key (sleutel toekennen aan een disk) en Lock (beveiligen van een file).

DOSAV2.0A is aangepast voor werken met KB-9, met de assembler van Carl Moser, en met Pascal. Binnenkort met FATE. Aan de hand van het voor fl. 10.== verkrijgbare artikel moet men zelf de controllerkaart met FDC 1770 ontwikkelen en bouwen (wire wrap of road-running). Het totale informatiepakket, waarin een complete source-listing met commentaar kost fl. 97,50. Overmaken op postrekening 841433 t.n.v. W.L. van Pelt te Krimoel a.d. IJssel. Of op de AMRO-bank aldaar t.p.v. bankrekening 44.11.06.471 t.n.v. W.L.v.Pelt.

ZAKBOEK 6502. DELFIA PRESS BV, Bob Bright. ISBN 906449 0287

Deze serie naslagwerken in zakformaat verschafft u beknopte informatie over en betrouwbare beschrijvingen van de voor-naamste kenmerken van de meeste gebruikte programmeertalen. Het Zakboek 6502 is een ideaal naslagwerk voor zowel beginnende gebruikers van 6502 microprocessors als voor professionele programmeurs die in de 6502-assembleertaal programma's willen schrijven.

HEX DUMP:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0400:	80	AA	80	80	8A	80	80	82	80	AA	AA	80	80	80	80	80
0410:	80	80	AE	AE	84	8B	8B	81	80	80	80	98	9D	84	B2	B7
0420:	84	80	81	80	8F	A0	84	98	A1	90	80	82	81	96	99	80
0430:	96	A4	84	82	81	81	80	8A	95	80	82	80	80	80	80	80
0440:	86	80	8A	90	80	80	82	80	82	A4	80	80	9A	80	82	80
0450:	80	A4	B5	84	9B	9F	91	80	81	80	80	94	80	83	97	81
0460:	80	80	80	80	80	80	80	A8	94	80	88	81	80	80	80	83
0470:	83	81	80	80	80	80	80	80	80	80	80	8A	85	80	80	80
0480:	84	98	81	80	80	80	80	96	A3	94	9D	81	95	82	83	80
0490:	88	95	80	80	95	80	80	81	80	86	83	94	98	83	80	83
04A0:	83	81	83	A3	85	80	83	94	83	83	80	95	AA	80	83	AB
04B0:	81	80	82	80	B7	B3	81	80	95	83	83	80	98	83	80	80
04C0:	97	83	94	82	82	83	80	83	A3	85	98	81	80	81	80	96
04D0:	83	94	96	83	94	82	83	80	96	83	94	82	A3	85	82	81
04E0:	80	A8	94	80	A8	94	80	80	80	A8	94	80	A8	94	80	80
04F0:	88	81	80	A0	86	80	89	90	80	80	82	80	B0	B0	90	80
0500:	8C	84	80	80	80	82	A4	80	80	98	81	82	80	80	86	83
0510:	94	80	86	80	80	81	80	96	B3	94	95	8D	85	82	83	82
0520:	96	83	94	97	83	95	81	80	81	97	83	94	97	83	94	83
0530:	83	80	96	83	84	95	80	90	82	83	80	97	83	94	95	80
0540:	95	83	83	80	97	83	81	97	83	80	83	83	81	97	83	81
0550:	97	83	80	81	80	80	96	83	84	95	83	95	82	83	81	95
0560:	80	95	97	83	95	81	80	81	82	97	80	80	95	80	82	83
0570:	80	80	80	95	90	80	95	82	83	80	95	98	81	97	A4	80
0580:	81	80	81	95	80	80	95	80	80	83	83	81	90	98	95	95
0590:	81	95	81	80	81	90	90	95	95	82	95	81	80	81	96	83
05A0:	94	95	80	95	82	83	80	97	83	94	97	83	80	81	80	80
05B0:	96	83	94	95	A4	85	82	81	81	97	83	94	97	A7	80	81
05C0:	80	81	96	83	84	92	83	94	82	83	80	83	97	81	80	95
05D0:	80	80	81	80	95	80	95	95	80	95	82	83	80	95	80	95
05E0:	A5	A0	85	80	81	80	95	80	95	85	A5	95	81	80	81	A5
05F0:	A0	85	98	89	90	81	80	81	95	80	95	82	96	80	80	81
0600:	80	83	A3	85	98	81	80	83	83	81	AA	83	80	AA	80	80
0610:	82	83	80	A4	80	80	80	89	90	80	80	80	82	A8	80	80
0620:	AA	80	82	83	80	98	9D	90	80	95	80	80	81	80	A0	84
0630:	80	8B	93	81	80	80	80	AA	85	80	80	81	80	80	80	80
0640:	A0	90	80	9C	AE	80	83	83	81	AA	B0	90	AA	80	95	83
0650:	83	80	80	B0	80	AA	80	81	80	83	81	A0	BA	80	95	AA
0660:	80	82	83	81	80	80	AA	8C	81	80	83	81	80	98	84	84
0670:	82	97	80	80	81	80	80	80	90	8A	90	85	88	88	81	AA
0680:	80	80	AA	80	95	82	80	81	80	84	80	80	95	80	80	81
0690:	80	80	88	80	80	AA	80	82	86	80	A8	80	90	AA	A6	80
06A0:	82	80	81	88	94	80	80	95	80	80	83	80	80	80	80	95
06B0:	95	95	81	81	A0	B0	80	AA	80	95	82	80	81	80	B0	80
06C0:	80	AA	80	95	80	83	80	A0	B0	80	AA	B0	85	8A	80	80
06D0:	A0	B0	80	A5	BA	80	80	8A	84	A0	A0	90	AA	81	80	82
06E0:	80	80	80	B0	90	82	8C	90	82	83	80	A0	B4	80	80	95
06F0:	90	80	82	80	90	A0	80	95	AA	80	82	83	81	90	80	90
0700:	A9	A8	81	80	81	80	90	80	90	95	94	95	82	82	80	B0
0710:	80	90	A0	97	80	81	82	81	A0	80	90	8A	B0	95	80	8C
0720:	85	A0	B0	90	A0	86	80	83	83	80	80	96	81	82	94	80
0730:	80	82	81	80	95	80	80	94	80	80	81	80	82	A9	80	80
0740:	A8	81	82	81	80	80	80	83	83	95	80	80	80	99	99	99
0750:	91	A6	A6	84	81	81	80	80	80	80	80	80	80	80	80	80
0760:																

TE KOOP AANGEBOREN

JUNIOR kast 19" + zware voeding, basiskaart + imeriaal, interfacekaart, 5 slot backpanel,
met EPROM's PM en TM samen: FL.150.==
Elektuur Erommer 2716/2732 met textool socket: FL. 75.==
8" FDOS controller kaart met FD 1771 FL. 90.==
2 x 8K Ram/Rom kaart met 2 x 8K Ram en in EPROM's FDOS 1.2 voor 8" FL.100.==
64K static Ram/Rom-kaart met 16K Ram FL. 75.==
Verkoop ten behoeve van een Belgisch lid.
Voor informatie of bezichtiging: Tel.: 01807 - 19881

ACORN ELEKTRON

Vele 6502 kenners onder ons kennen al enige tijd de hoog aangeschreven BBC computer. Een goedkoper broertje daarvan issoireerde Simon Voortman tot het verstrekken van de volgende informatie.

De Acorn Electron is een nog betrekkelijk nieuw ontwerp (uit 1983). De Electron is uitgerust met dezelfde (beroemde) Basic van de BBC. 't Is eigenlijk een kruising tussen Basic en Pascal. Hierdoor komen in deze Basic-versie bijv. PROCEDURES en lokale variabelen voor. Een procedure (afkorting: PROC) is eigenlijk gewoon een subroutine, alleen behoeft er geen GOSUB gebruikt te worden! In plaats van bijvoorbeeld GOSUB 500 zeg je dan bijv. PROC main-program. De computer zoekt dan zelf de subroutine op: deze moet dan beginnen met: DEF PROC main-program. In plaats van RETURN schrijf je dan: ENDPROC. Op deze manier hoeft je geen sarongadressen bij te houden, en de procedures kunnen overal in het programma aangeroezen (en ook geplaatst) worden.

Bij de Electron wordt ook een flink dik (290 pagina's) handboek geleverd, samen met een programmeerboek (Start Programming with the Electron) en een Introductory cassette, met daarop best interessante programma's, die de grafische en geluidsmogelijkheden goed illustreren.

De Electron heeft een 6502 A 2 MHz processor, en dit is vooral bij listen van programma's goed te merken: het programma vliegt over het scherm. Tevens zijn de cursor en de kleuren in elke mode te (her)programmeren. Al dit moois vraagt ook zijn prijs. In de laagste (beeldscherm)modus vraagt het beeldschermgeheugen zo'n 20K RAM van de totale 32K geheugen. De Electron heeft wat minder standaard interfaces. de BBC is dan ook 2 x duurder. Er zijn aparte aansluitingen voor TV, RGB-monitor en een uitgang met een composite video signaal. Voor het geluid zorgt een ingebouwde luidspreker. Op de Electron kunnen ook, echter niet alle BBC-programma's draaien.

COMMODORE BASIC 3.5 (As present in the Commodore C-16).

Tokenised Microsoft BASIC keywords and addresses M. de Vries
 Analogous to the publication for Commodore-64 by A. Mueller.
 DE 6502 KENNER, December 1983, pages 5-8.

COMMANDS

KEYWORDS CORRESPONDING TO 818E
 ADDRESSES CORRESPONDING TO 8383
 THE ADDRESSES OF ROUTINES FOR COMMANDS ARE THE ADDRESSES
 MINUS 1, BECAUSE THE ROUTINES ARE INVOKED THROUGH RTS.

KEYWORD	TOKEN	ADDR-1	KEYWORD	TOKEN	ADDR-1
END	80	8CD9	RESUME	D6	B43F
FOR	81	ADC9	TRAP	D7	B42A
NEXT	82	9293	TRON	D8	B651
DATA	83	8DAF	TROFF	D9	B654
INPUT#	84	90ED	SOUND	DA	B848
INPUT	85	9107	VOL	DB	B8BC
DIM	86	969A	AUTO	DC	B6CC
READ	87	914E	PUDEF	DD	B543
LET	88	8E7B	GRAPHIC	DE	C5C2
GOTO	89	8D4C	PAINT	DF	B8D8
RUN	8A	8BBB	CHAR	E0	B9D3
IF	8B	8DE0	BOX	E1	BAE1
RESTORE	8C	8C99	CIRCLE	E2	C01D
GOSUB	8D	8D2B	GSHAPE	E3	BD34
RETURN	8E	8D82	SSHAPE	E4	BE28
REM	8F	8E0A	DRAW	E5	C4D8
STOP	90	8CD7	LOCATE	E6	C50E
ON	91	8E1A	COLOR	E7	C519
WAIT	92	9E69	SCNCLR	E8	C566
LOAD	93	A7F2	SCALE	E9	C5B7
SAVE	94	A7DD	HELP	EA	B6E7
VERIFY	95	A7EF	DO	EB	B556
DEF	96	9A9C	LOOP	EC	B602
POKE	97	9E11	EXIT	ED	B5AB
PRINT#	98	8FDF	DIRECTORY	EE	C8BB
PRINT	99	8FFF	DSAVE	EF	C940
CONT	9A	8D02	DLOAD	F0	C950
LIST	9B	8AFE	HEADER	F1	C967
CLR	9C	8A97	SCRATCH	F2	C99B
CMD	9D	8FE5	COLLECT	F3	C9CB
SYS	9E	A7B4	COPY	F4	C9D9
OPEN	9F	A84C	RENAME	F5	C9F3
CLOSE	A0	A859	BACKUP	F6	C9FF
GET	A1	90B7	DELETE	F7	AE59
NEW	A2	8A78	RENUMBER	F8	AB8E
GO	CB	(8C6B)	KEY	F9	B728
ELSE	D5	8E0A	MONITOR	FA	FF51

- Note: 1) Bit 7 in the last character of each keyword is set to determine the end of a keyword.
 2) Keywords are tokenised by adding \$80 to their relative (hex) position in the table.

COMMODORE BASIC 3.5 (As present in the Commodore C-16).

Tokenised Microsoft BASIC keywords and addresses

MISCELLANEOUS KEYWORDS

KEYWORDS CORRESPONDING TO 8219 AND 8373 (*)

KEYWORD TOKEN

TAB(A3	NOT	A8
TO	A4	STEP	A9
FN	A5	USING	* FB
SPC(A6	UNTIL	* FC
THEN	A7	WHILE	* FD

DYADIC OPERATORS

KEYWORDS CORRESPONDING TO 8230
PRIORITIES AND ADDRESSES CORRESPONDING TO 8453

THE ADDRESSES OF ROUTINES FOR DYADIC OPERATORS ARE THE
ADDRESS MINUS 1, BECAUSE THE ROUTINES ARE INVOKED THROUGH
A RTS INSTRUCTION.

KEYWORD TOKEN ADDR-1 PRTY

+	AA	9E9D	79	addition
-	AB	9E86	79	subtraction
*	AC	A07A	7B	multiplication
/	AD	A196	7B	division
↑	AE	A5ED	7F	exponentiation
AND	AF	95FA	50	logical AND
OR	B0	95F7	46	logical OR
monadic "-"	AB	A626	7D	negation
monadic NOT	AB	9464	5A	logical NOT
>	B1	9627	64	comparison
=	B2	9627	64	comparison
<	B3	9627	64	comparison

- Note: 1) Bit 7 in the last character of each keyword is set to determine the end of a keyword.
2) Keywords are tokenised by adding \$80 to their relative (hex) position in the table.

COMMODORE BASIC 3.5 (As present in the Commodore C-16).

Tokenised Microsoft BASIC keywords and addresses

```
-----
: FUNCTIONS
:
: FUNCTIONS CORRESPONDING TO 823D AND 828D (*)
: ADDRESSES CORRESPONDING TO 8415
:
```

KEYWORD	TOKEN	ADDR	KEYWORD	TOKEN	ADDR
SGN	B4	A2BE	STR\$	C4	9B66
INT	B5	A358	VAL	C5	9D93
ABS	B6	A2DD	ASC	C6	9D78
USR	B7	0580	CHR\$	C7	9DBB
FRE	B8	9A62	LEFT\$	C8	9CCF
POS	B9	9A7D	RIGHT\$	C9	9D03
SQR	BA	A5E4	MID\$	CA	9D15
RND	BB	A787	RGR	* CC	BF79
LOG	BC	A81E	RCLR	* CD	BF85
EXP	BD	A660	RLUM	* CE	BF87
COS	BE	AA70	JOY	* CF	BFC1
SIN	BF	AA77	RDOT	* D0	BFFD
TAN	C0	AAC0	DEC	* D1	9E1B
ATN	C1	AB1A	HEX\$	* D2	B587
PEEK	C2	9DFA	ERR\$	* D3	B4BE
LEN	C3	9D61	INSTR	* D4	(B386)

- Note: 1) Bit 7 in the last character of each keyword is set to determine the end of a keyword.
 2) Keywords are tokenised by adding \$80 to their relative (hex) position in the table.

BOEKBESPREKING

Van Peter van Harten ontvingen we de volgende informatie:

FLITSEND FORTH door Alan Winfield
 Uitgeverij: Academic Service, Den Haag.
 ISBN 90-6233-115-7 Prijs: ca. Fl. 35.==

Het boek is een vertaling van "The Complete FORTH. A new way to program computers."

De eerste helft is een cursus FORTH. In de tweede helft worden speciale FORTH-technieken getoond. Een FORTH-79 overzichtskaart wordt meegeleverd. Het is een helder geschreven boek, geschikt voor beginners en gevorderden. In de voorbeelden worden BASIC-equivalenten van FORTH-constructies getoond en er zijn oefeningen met uitwerkingen.

INLEZEN SOURCE Micro-ADE

Gebruikers van de Assembler/Disassembler/Editor Micro-ADE hebben inmiddels al wel ontdekt dat enige adressen vanaf \$2EA3 belangrijk zijn voor de I/O-aktiviteiten en het bepalen van de werkvelden, w.o. de source-buffer, de symbol table, de XREF-table. Een enkele keer komt het voor dat men in de veronderstelling verkeert dat de eigen tapes met Micro-ADE source niet zonder meer kunnen worden ingelezen door een andere gebruiker die hetzelfde formaat toe oast. Men gaat er dan vanuit dat bij de ander het adres van de source-buffer moet overeenstemmen met die van de oorspronkelijke gebruiker. Dit is ten onrechte. Het kan geen kwaad, maar het is overbodig! Micro-ADE leest het programma van tape en olaat het in de source-buffer die door de ontvanger reeds eerder is bepaald. Heeft men dus gedumpt op tape vanaf \$4600 en heeft de ander de source-buffer gedefinieerd op adres \$5000, dan leest Micro-ADE zonder haderen in naar adres \$5000. Gemak dient de mens!



all-round

All-round Systemen Nederland B.V., Stationsweg 23, 2182 BA Hillegom
postbus 212, 2180 AE Hillegom. tel. 02520-19544. telex 41443 allro nl.

WIJ LEVEREN SNEL EN TEGEN ZEER GUNSTIGE PRIJZEN:

Diskettes

van verschillende merken zowel 8" als 5 $\frac{1}{4}$ ". Tevens formatted en cleaning diskettes.

Papier

voor kleingebruikers leveren wij dozen van 1000 vel 38 cm.

Inktlinten

linten en kassettes voor alle gangbare printers.

Computermeubilair

o.a. ideale tafeltjes om een compleet micro systeem op te plaatsen.

Data-safe's

een mini safe voor 80 x 5 $\frac{1}{4}$ ", of 30 x 8" diskettes.

Opbergsystemen

o.a. voor diskettes, tapes, kassettes en alle maten output.

Geluiddempende kappen

brengen het geluid van uw printer 75 tot 90% terug.

Alles voor uw computer



all-round

Inhoudsopgave

Pagina 2

Number	Page	Artikel	Writer	System	Language
22	41	Ophalen tekst	W.L.van Pelt	Junior	Assembler
22	43	Locate en replace	H.K.Hemminga	Kim	Assembler
22	47	Basicode	J.F.van Toledo	Junior	Assembler
22	51	Aanpassingen basic VN V2.1	- - -	Junior	Assembler
22	56	Automatisch opstarten	J.F.van Sprang	Junior	Assembler
22	56	Timer en tijdstijftechniek voor de 6522	J.P.van Toledo	Junior	Assembler
22	59	OKI printer aan VBA	F.Smeehuyzen	Algemeen	Assembler
22	60	OKI printer met de Junior	A.S.Hankel	Kim	Assembler
22	62	Easy editing supporter	R.U.Vissers	Junior	Assembler
22	63	Simple geheugen uitbreiding	H.Burgers	Algemeen	Hardware
22	63	Simple step debug programma	M.A.v.d.Laan	Junior	Assembler
22	65	tiny basic voor de Junior	F.van Kenhove	Junior	Assembler
22	66	Floppy voor Junior, Kim, Sym en Aia 65	A.Muller	Algemeen	Assembler
22	67	Lotto getallen	K.van Nieuwenhove	Algemeen	Basic
22	68	Frequentieteller met de 6522	J.Susan	Junior	Assembler
22	69	Aanpassingen basic VN V2.1 voor Junior	K.van Nieuwenhove	Junior	Assembler
22	70	Salontemperatuur met de Aia 65	T.Schouten	Aia 65	Assembler
22	70	Beschrijving Photon floppy system	- - -	Algemeen	- - -
22	70	Uitbreidingen	F.G.Jonkman	Kim	Assembler
22	71	Bitmask programma	R.Vissers	Junior	Assembler
22	72	Relais en relais	H.W.v.d.Linden	Acorn	Basic
22	72	Aanpassingen aan basisc VN V2.1	H.Guijpers	Junior	Hardware
22	72	Basic processor compressor	K.van Nieuwenhove	Junior	Assembler
22	73	Gebitmask voor de VBA 6522	R.Uphoff	Algemeen	Assembler
22	74	Basic data regels genereren	A.Kuitens	Algemeen	Algemeen
22	74	Spiraal	H.van Hintum	Algemeen	Basic
22	74	Basic voor de Junior	H.W.v.d.Hartog	Kim	Assembler
22	75	Reset op start	E.J.Melisschedijk	Junior	Assembler
22	75	Service rapporten bestand	E.J.Melisschedijk	Junior	Assembler
22	76	Van 6502 naar 65 convertie	K.R.van Hintum	Junior	Basic
22	76	Makkelijk van Junior een Senior	F.Praanssen	Junior	Assembler
22	76	VIS	J.Cristen	Algemeen	Basic
22	76	Printen op GKI Microline met de Junior	W.L.van Felt	Junior	Assembler
22	77	Seven segments I/O	H.W.v.d.Hartog	Kim	Assembler
22	77	Hexadecimaal optellen en aftrekken	R.Smeehuyzen	Kim	Assembler
22	78	Segmentprogrammae	R.Langeveld	Junior	Assembler
22	78	Address	J.F.van Toledo	Junior	Assembler
22	79	Mastercard op beeldscherm voor Junior	R.Vissers	Junior	Assembler
22	79	Digitale klok met 8 sanslubbare poorten	J.H.Vernimmen	Kim	Assembler
22	80	Programmaarbeider voor Wetenschap	D.Gackx	Algemeen	Assembler
22	80	Junior/Pakbon	F.Smeehuyzen	Algemeen	Basic
22	81	Programmeren met Basic	J.Janssen	Algemeen	- - -
22	81	CBM centronics interface	A.Muller	CBM 64	Assembler
22	82	Nogmaals Basic compressor	R.Uphoff	Algemeen	Assembler
22	82	Input an expression	R.Uphoff	Hippie II	Assembler
22	83	Koppeling van Junior aan 8" flopp	K.van Nieuwenhove	Junior	Algemeen
22	83	Scan	R.A.F.Bens	Junior	Assembler
22	84	Start-up	A.Hankel	Algemeen	Basic
22	84	tekstverwerking met teletype	J.C.J.Beijer	Kim	Assembler
22	85	telefoonbeepcode	J.P.van Toledo	Junior	Assembler
22	85	NET schakeling voor Junior	E.van den Driessche	Kim	Assembler
22	86	Automatische registratiedeling II	A.A.Zwart	Junior	Assembler
22	86	Patches op Micro 404	P.van Niekerk	Junior	Assembler
22	87	Een floppy-disk van de Aia 65	A.S.Hankel	Junior	Assembler
22	87	Using 8080 with the 6522	E.A.Gerritsen	Aia 65	Hardware
22	88	Microbasic Basic routines	S.van Roekel	Algemeen	forth
22	88	Galois	R.Uphoff	Algemeen	- - -
22	89	Micro Ade error display	R.Uphoff	Junior	Assembler
22	89	Hexdump	R.Uphoff	Junior	Assembler
22	90	Commentaar op Factor/Pakbon	R.J.M.Smeehuijzen	Junior	Assembler
22	90	Portret maken op C-plot	R.Uphoff	Algemeen	Basic
22	91	Uitbreidende 6502 disassemlbler	R.Uphoff	Junior	Assembler
22	91	CBM 64 characterized Basic keywords	C.van Roekel	Algemeen	Basic
22	92	Muziekbestand	A.Muller	CBM 64	- - -
22	92	Schuiven of springen	J.P.van Toledo	Algemeen	Basic
22	93	Uitbreidingen voor de CBM basic	R.de Freetere	Junior	Assembler
22	93	TSK verwerker triaden	F.J.M.Smeehuijzen	Junior	Assembler
22	94	Hexadecimaal omzetter	H.A.van de Laan	Junior	Assembler
22	94	clock on display and screen	D.Blok	Junior	Assembler
22	95	Juniors hex monitor en Acorn system	R.Jonk	Junior	Assembler
22	96	Annuiteitenhypothek	A.van der Meutter	Acorn	Assembler
22	97	Recorder/telex aan CBM 64	F.Jonkman	Junior	Basic
22	98	PRCHA	F.Weber	CBM 64	- - -
22	99	Programmeren van 2764	F.Lopez	Junior	Assembler
22	10	Bouw eerst een Miljonair	R.Langeveld	Algemeen	- - -
22	11	Bugs in Super star trek	J.H.Vernimmen	Junior	- - -
22	12	Program Cassette 1	R.Banen	Algemeen	Basic
22	13	Schuiven of springen deel 2	A.Brouwer	Junior	Assembler
22	14	Leds op Junior	R.de Preetere	Junior	Assembler
22	15	Printer routines voor STAR DP 510	H.Buerman	Junior	Assembler
22	16	Uitbreiding Sym basic voor Junior	S.I.Woldringh	Algemeen	Assembler
22	17	Wiskunde programma	F.J.M.Smeehuijzen	Junior	Assembler
22	18	Lichtkrant/beeldscherm	- - -	Algemeen	Basic
22	19	Horse tutor	I.van Rijsei	Junior	Assembler
22	20	Uitbreiding Sym basic voor Junior	T.G.de Driever	Junior	Assembler
22	21	Obsessie	F.J.M.Smeehuijzen	Junior	Assembler
22	22	Processor met ...Push X	H.Burgers	Algemeen	- - -
22	23	Tips & Tricks	- - -	Algemeen	Basic
22	24	De VDU kaart	J.J.Janssen	Junior	Assembler
22	25	Program Casette 2	A.Brouwer	Junior	Assembler